AFRL-HE-WP-TR-2006-0027

AIR FORCE RESEARCH LABORATORY

ENHANCED NIGHT VISION VIA A COMBINATION OF POISSON INTERPOLATION AND MACHINE LEARNING

Leonard McMillan
Wei Wang

University of North Carolina
Department of Computer Science
Chapel Hill NC 27599-3175

FEBRUARY 2006

FINAL REPORT FOR 16 SEPTEMBER 2004 TO 15 DECEMBER 2005

Human Effectiveness Directorate
Warfighter Interface Division
Wright-Patterson AFB OH 45433

NOTICE

This report was cleared for public release by the Air Force Research Laboratory Wright Site (AFRL/WS) Public Affairs Office (PAO) and is releasable to the National Technical Information Service (NTIS). It will be available to the general public, including foreign nationals.
[PAO Case Number: AFRL/WS-06-0800, 29 March 2006.]

This report is releasable to the National Technical Information Service (NTIS).

This technical report has been reviewed and is approved for publication.

FOR THE DIRECTOR

/S/

MARIS M. VIKMANIS, DR-IV
Chief
Warfighter Interface Division
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
February 2006	FINAL	16 Sep 2004–15 Dec 2005

4. TITLE AND SUBTITLE

Enhanced Night Vision Via a Combination of Poisson Interpolation and Machine Learning

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA8650-04-2-6543

5c. PROGRAM ELEMENT NUMBER
62301E

6. AUTHOR(S)

Leonard McMillan and Wei Wang

5d. PROJECT NUMBER
7184

5e. TASK NUMBER
11

5f. WORK UNIT NUMBER
33

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of North Carolina
Department of Computer Science
Chapel Hill NC 27599-3175

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Materiel Command
Air Force Research Laboratory
Human Effectiveness Directorate
Warfighter Interface Division
Wright-Patterson Air Force Base OH 45433-7022

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/HECV

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-HE-WP-TR-2006-0027

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited. (PAO Case Number: AFRL/WS-06-0800, 29 March 2006)

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Our research initiative focuses on enhancing semiconductor-based night-vision imagery via computation. The fundamental problems of imaging under low-light conditions are noise suppression, overcoming low image contrast, and the loss of important perceptual cues. We have developed three new image-processing techniques to address these problems. These include non-linear spatio-temporal denoising filters, variants of Poisson integration for localized-contrast adjustment, and machine-learning methods for reintroducing perceptual cues. The goal of our work is to enhance night-vision imagery by improving its sensitivity, and aiding in its rapid and accurate interpretation. We have applied our methods to low-light visible, near infrared (NIR), and short-wave infrared images (SWIR). In this annual report on the first phase of our research, we describe and evaluate the processing methods that we have developed to date. Our objective in this phase was to demonstrate the feasibility and potential of these new computational approaches.

15. SUBJECT TERMS

Night-Vision, Noise Reduction, Colorization, Tone Mapping, Machine Learning, Poisson Interpolation, Adaptive Filters, Belief Propagation, SWIR, Image Fusion

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
UNCLASSIFIED					Dr. Darrel G. Hopper
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(include area code)*
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR	118	(937) 255-8822

This page intentionally left blank.

CONTENTS

LIST OF FIGURE CAPTIONS

LIST OF TABLE TITLES

FOREWORD

This DARPA-funded AFRL-managed Cooperative Agreement FA8650-04-2-6543 was awarded under Defense Advanced Research Projects Agency (DARPA) Broad Agency Announcement (BAA) No. 04-17 dated 8 March 2004 entitled "Innovative Information Exploitation Technology and Systems," Part A.1.4. "Battlespace Visualization," http://www.darpa.mil/baa/baa04-17.htm, to the University of North Carolina (UNC) against their proposal number P-0417-100750 dated 18 Jun 04, entitled "Enhanced Night-Vision Via a Combination of Poisson Interpolation and Machine Learning." Cooperative Agreement FA8650-04-2-6543 was awarded 16 September 2004 to UNC in Chapel Hill for a 15-month seedling effort (12 technical, 3 reporting) funded in the amount of $240,648 provided by DARPA to AFRL via ARPA Order No. T212/00. Deliverables are reviews, demonstrations, algorithms, and reports (quarterly, final). Formal reviews were conducted on 7 October 2004 at UNC, 13 December 2005 at DARPA, 25 January 2005 at DARPA, 15 March 2005 at WPAFB, and 1 June 2005 at UNC. The final review is scheduled for 20 October 2005 at DARPA. The 3 month reporting period comprises preparation and submission by UNC of a draft final report by 15 October 2005, government comments by 15 November 2005, and revision and submission by UNC of the approved final report by 15 December 2005.

This report has been formatted in accordance with a commercial standard, with tailoring from the AFRL Scientific Technical Information Office. This standard is as follows: "Scientific and Technical Reports—Elements, Organization, and Design," American National Standard ANSI/NISO Z39.18-1995 (NISO Press, Bethesda MD, 1995), which is available electronically via the following website address: http://www.wrs.afrl.af.mil/library/sti-pubh.htm
The Government Program Manager for this DARPA-funded AFRL-managed effort was Dr. Darrel G. Hopper of AFRL, who accomplished the technical review of this document.

PREFACE

The DARPA Information Exploitation Office (DARPA/IXO) solicited under BAA04-17 for proposals for advanced research and development of enabling technology, critical subsystems, and full system concepts that would provide revolutionary improvement to the efficiency and effectiveness of military command, control, intelligence, surveillance, reconnaissance (C2ISR) and strike operations in complex battlespaces. The goal was to identify and develop novel ideas about sensing, signal processing, target characterization, data fusion, target tracking, predictive awareness, battle management, collaborative planning, and visualization that can contribute to future warfighting effectiveness. Ideas were to address 1) ways to employ new scientific/technical developments to achieve significant increases in component performance; 2) novel combinations of existing technologies into systems that create new warfighting capabilities; or 3) combinations of both. Proposed efforts were to constitute at least two phases. The first phase was to seek to establish the technical feasibility of the concept, in an exploratory effort lasting between 3 months and one year. Phases beyond the first were to be proposed as options leading toward increasingly robust technology or system development, and consist of 12-18 month segments with specific capability goals to be achieved at the end of each segment. "Battlespace Visualization" was a specific Area of Interest under the BAA which called for new concepts and innovative technologies for advanced visualization and presentation, immersive interfaces, advanced human computer interfaces, natural language, gesture and mixed modal interfaces and other capabilities that may facilitate accurate and rapid assimilation and manipulation of complex, multi-source battlefield data by strategic and tactical commanders.

The objective of this Year 1, 12-month seedling effort by UNC was to initiate development of novel methods for constructing night-vision images with a day lit appearance to provide warfighters with daylight quality night-vision capability The specific goals of this effort included investigation of the feasibility of applying Poisson interpolation to the problem of enhancing night-vision (NV) images and evaluation of the viability of machine-learning approaches to the problems of (a) estimating daytime reflectance from local texture information and (b) combining gradient images from multiple NV sensors to estimate daytime gradients. Additional goals were (c) to initiate development of fast and efficient algorithms for Poisson interpolation, multispectral (MS) recognition, and classification and (d) to plan work on next generation NV capabilities which will provide imagery that will in many ways surpass daylight images in both information content and quality.

ACKNOWLEDGEMENTS

We gratefully acknowledge the financial support received from DARPA and program/contact management support from the AFRL. We thank Dr. Darrel G. Hopper, Ms. Christine Task, and Dr. Alan Pinkus of AFRL, Dr. Theodore Bially and Mr. Jeffrey Paul of DARPA, and Mr. Bob Schulte of Solers Inc. (DARPA SETA Contractor) for their keen technical insights and guidance of this effort on behalf of the government.

This page intentionally left blank.

1. SUMMARY

The capability to work in complete darkness has been, for the past 20 years, one of the United State's primary tactical advantages in military operations. It has been suggested that this advantage is rapidly eroding with the availability of commercial and foreign-produced night-vision goggles.

Our research explores new approaches to night-vision imaging designed to help maintain the U.S. military's tactical advantage in night-vision for the next several decades. By leveraging abundant computational resources, multi-spectral, intensified image sensing, and recent advances in image enhancement, we demonstrate night-vision capabilities that approach daytime viewing. In the next 10 years, it might be possible to construct night-vision systems with this and other capabilities such as hyper-acuity, including super-resolution and high-dynamic range using our approach. Our research combines novel approaches to sensing, signal processing, data fusion, and visualization. It employs computation to achieve significantly increased sensitivity from existing components.

We envision a new generation of night-vision systems that provide imagery more easily interpreted by soldiers. Such systems would provide enhanced contrast and improved depth perception, and would significantly aid in the accurate interpretation of battlefield situations.

We have developed a suite of new image-processing methods for enhancing night-vision image. These include:

- Noise reduction methods using non-linear spatio-temporal filters
- Contrast manipulation using Poisson interpolation
- Image enhancements using machine-learning techniques

In the first phase of this research effort, we have developed and evaluated image processing methods specifically targeting low-light and night-vision video. We have also applied our methods to registered and unregistered multi-spectral imagery.

Technical Approach:
Our approach is to create a virtual imaging array in which every pixel has its own independent exposure time, which is set according to the ambient light level falling onto its, and its nearby neighbors', sensors. The exposure level, in our approach, adjusts adaptively to obtain the desired signal-to-noise ratio and image contrast. This contrast adjustment considers both the temporal and spatial neighborhoods and noise levels of each pixel, or photosite. This allows the illumination-to-luminance mapping function to adapt dynamically over the entire image, thus enhancing contrast and brightness levels, while maintaining a perceptually faithful image. We also map the resulting unlit images to simulate a more interpretable illumination condition. The key components to our approach are non-linear spatio-temporal filtering for noise suppression, Poisson integration for contrast manipulation, and machine learning for reillumination. Currently, all of the methods are computationally intensive. We are optimistic that a combination of technology advancements and algorithm improvements can make these methods viable for night-vision systems in the next 10 years.

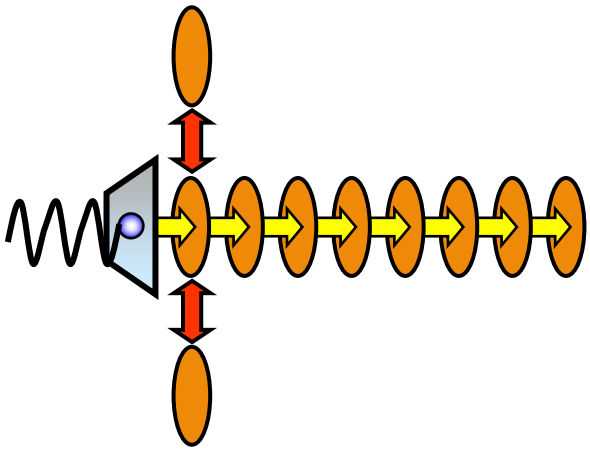


Figure 1: Illustration of the processing of photons as they hit each photosite on the sensor. The photosite (gray) captures at a fixed high speed rate. This sample is compared with its spatial neighbors (shown here above and below) to determine how to best enhance contrast. This contrast is enhanced by combining sample information from the delay line (shown here horizontally) of samples from that photosite at earlier times. If necessary, further filtering can occur spatially, especially in areas of motion.

Comparison with Current Technology:

Standard night vision systems are essentially analog devices. They employ low-noise image intensifiers (photomultiplier tubes) directly coupled to Cathode Ray Tubes CRTs for display. Non-linear "gamma" mapping functions are used most frequently to adjust image contrast. Non-linear gamma mappings are global functions of pixel intensity only. They do not consider local illumination variations, thus they can only adjust contrast and brightness over the entire image, and subsequent processing of a gamma corrected image is ill-defined conceptually. Our methods locally adapt both brightness and contrast over the entire image plane, thus providing the most detail possible, as well as a perceptually consistent result. In other words, despite dramatic enhancements in both brightness and contrast, our processing results in easily interpretable images that do not appear washed out; nor do they loose details in overexposed and underexposed image regions.

Today's multi-spectral and wideband IR night-vision systems employ solid-state CCD detectors. These systems frequently employ analog image processing, and thus are limited to simple methods for noise reduction and contrast adjustment. Advanced experimental night-vision systems employ digital processing pipelines. These systems are compatible with our approach. However, we have no knowledge of anyone else using the methods we are exploring for night-vision applications. We believe that this is for three primary reasons: the methods we propose are very computationally demanding, they require considerable local storage, and they rely on iterative and non-linear processing techniques. They have only recently been used for off-line image processing applications.

2. INTRODUCTION

We propose an innovative approach for maintaining the U.S. military's supremacy in night-vision for the next several decades. By leveraging abundant computation resources, multi-spectral and intensified image sensing, and recent advances in image enhancement, we propose to reconstruct night-vision images that rival daytime views. This report presents a novel approach to sensing, signal processing, data fusion, and visualization and it addresses new scientific and technical developments that will achieve significant increases in existing component performance.

Night-vision systems are one of the few remaining analog technologies in wide military use. They employ low-noise image intensifiers (photomultiplier tubes) directly coupled to Cathode Ray Tubes CRTs for display. However, the transition to solid-state imaging devices is imminent and it introduces new challenges an offers new opportunities. The challenges include different spectral sensitivities and different noise characteristics. Solid-state imagers, unlike night-vision scopes, also rely on image plane sampling, and a constant scan out. Solid-state imaging devices also offer the potential of inserting processing between the imaging and display components. This opens the door to a vast array of possibilities.

We envision a new generation of night-vision systems that provide imagery more easily interpreted by soldiers. Such systems would provide enhanced contrast and improved depth perception, and would significantly aid in the accurate interpretation of battlefield situations. Our approach combines the low-level recognition capabilities of domain-specific machine learning with new methods for interpolating images. The degree of enhancement is user controllable, varying all the way from raw unprocessed night-vision images to highly augmented daylight approximations.

We have developed and evaluated night-vision enhancement methods based on recent innovations in non-linear image filtering, Poisson integration, and machine learning. Our algorithm variants are optimized for the unique problems of low-light imaging and they take advantage of the increased flexibility of solid-state devices. We exploit the frame-sequential nature of solid-state imagers to enable noise-removal using temporal continuity. We also employ temporal information to enable spatially varying contrast adaptation via a per-pixel gain factor.

Nevertheless, there are still fundamental problems that hinder the interpretation of noise-free contrast adjusted night-images. These difficulties are due to a lack of important perceptual cues in night vision imagery. A human's ability to judge shape and identify objects depends on a wide variety of features that are either absent in, or inconsistent with, night-imagery. This is particularly problematic when examining non-visible spectrum (NIR or SWIR). Examples of such features include the lack of shading gradients due to illumination, and the prevalence of self-emission in the infrared imagers. In the case of night-vision, these desired features must be derived from the local context and known variations in the response of particular sensors (ex. visible light, intensified, NIR, and SWIR) to particular materials. The domain of military applications is suitably constrained (i.e. it is known a priori whether a given contact will take place in urban, jungle, or desert terrain) that extensive sets of training data (daylight example

images) could be acquired and used by machine-learning methods. We employ machine-learning methods based on the existence of such ground-truth training data. Using this approach, we have developed domain-specific models for supplying *plausible* estimates of missing data, which are often critical discrimination cues.

In this first phase of our research, we address potential capabilities, and ignore issues of practicality and computational complexity. Our goal is to demonstrate the range of possibilities enabled by computationally augmented night-vision, but we do not consider how such capabilities might be packaged and computed in real-time. Our working assumption is that computation is essentially free. In future work we expect to develop more time and resource efficient algorithms and approaches.

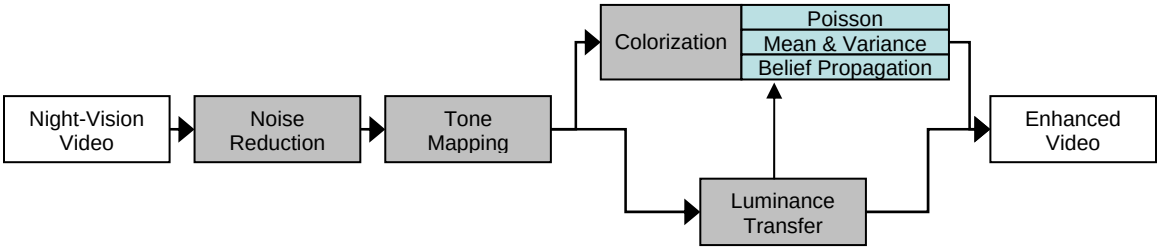


Figure 2: Pipeline for night-vision video enhancement using our combination of noise reduction, tone mapping, and a choice of luminance transfer and/or colorization. Noise reduction cleans up the dark footage, while tone mapping maps its over a more appropriate dynamic range. Luminance transfer can account for differences in relative luminances in non-visual spectra. Colorization can then reconstruct colors using a variety of techniques.

We break down our process of computationally enhancing night vision according to the simplified pipeline depicted in Figure 2. We treat each stage independently; however, there are several interactions and possible feedback paths. In general, our system assumes a video stream— a consecutive sequence of images with uniform exposure and gain settings unless otherwise specified. We also concentrate primarily on the case when the video stream is in its steady state, and treat start-up issues separately.

In the first step of our process, we apply noise reduction to the incoming source images. The signal-to-noise ratios (SNR) of solid-state imaging systems are dramatically reduced under low-light conditions. There are a variety of possible noise sources in solid-state sensors that confound imaging in low-light situations; these include readout, photon shot, dark current, and fixed pattern noise in addition to photon response non-uniformities [Reibel et al 03]. We assume that dark-current noise and fixed-pattern noise can be removed via subtraction of a reference dark images at the same temperature and exposure setting. Photon shot and readout noise are our primary problems, but we assume they are zero-mean, so if we can get multiple samples of the same pixel from temporally adjacent frames, we can average out the error. A significant problem in dealing with dark areas captured with CCDs is that the amplitude of sensor read noise is independent of exposure whereas photon shot noise varies linearly with exposure time. Read noise is more significant than shot noise in low-light conditions. Thus, under low-light conditions the SNR is comparatively smaller than the reported specifications.

We have developed an adaptive non-linear spatio-temporal filter to be used for noise reduction. This approach exploits both temporal and spatial continuity to cancel-out uncorrelated zero-mean noise sources. Our filter is non-homogenous and it adapts both its temporal and spatial extents depending on the signal content. We have also evaluated both causal and non-causal variants (non-causal versions are practical in cases where viewing latency is permissible).

The second stage of our processing pipeline is called tone-mapping. The objective of tone-mapping is to maximize the local contrast while maintaining an overall relative luminance that is faithful to the observed scene. The reason that employ local contrast adaptation rather than global (over the entire image) contrast and gain adjustment is that the human visual system is more sensitive to contrast than the local illumination (For example, a given grey value of 128 may be perceived as dark against the sky, yet bright within a shadowed region). We exploit this characteristic of the visual system to reuse and remap intensity levels to optimize the perceived image content.

Our tone mapping approach considers that SNR varies with luminance level. Thus, details in dark regions are less accurate than those in brighter regions. A tone mapper specialized for low-light and night-vision video should therefore associate a confidence level for details based on their luminous intensity. For instance, in the brightest areas of a video where the CCD received a reasonable exposure, the mix of details and large-scale features should be adjusted to achieve the tone mapping objectives. In darker areas the details should be attenuated to suppress noise.

The denoising and tone-mapping stages can be applied to both visible and non-visible image sources. However, there are several alternative third-stage image-enhancement stages whose application depends on the source image type. One of the problems with low-light visible spectrum images is the lack of useful color information. Color is an important perceptual cue for many classification and identification tasks. We employ a variety of machine-learning methods to colorize images based on statistical sampling of training sets, reintegration of sparse color data using Poisson methods, and an iterative multi-hypothesis resolution method called belief-propagation. The details of all processing stages are described in Section 3.

In Section 3.6 of this report we evaluate the results of our enhancement methods. We illustrate each of the methods on low-light visible spectrum, Near-Infrared, and Short-Wave Infrared sensors where applicable. We provide comparisons with alternative methods where they exist. In Section 4 we discuss the results in terms of our original project goals. In addition to the printed document, all results and software are also provided on an accompanying DVD-ROM.

We conclude with a discussion of future potential algorithmic enhancements to further enhance night-vision imaging systems. We also discuss the current algorithm complexities of the night-vision enhancement methods that we have developed and the potential that these methods could be sped up via dedicated hardware, and/or more efficient algorithms. Finally, we talk about possible partnering strategies and dissemination channels for the outcomes of the first phase of this research grant.

3. METHODOLOGY AND RESULTS

The three core objectives of this research were to develop:

- Noise reduction methods using non-linear spatio-temporal filters
- Contrast manipulation using Poisson interpolation
- Image enhancements using machine-learning techniques

In the following sections, we present a series of algorithms which achieve these goals. First is the ASTA (Adaptive Spatio-Temporal Accumulation) noise reduction filter, which directly addresses the sensor noise issue. The issue of contrast enhancement is first addressed here via a non-iterative tone mapping algorithm which works with ASTA to enhance the noise-reduced footage.

We then address colorization and contrast manipulation using Poisson methods and machine-learning techniques. Via Poisson interpolation we make nighttime luminances resemble daytime luminances and then colorize them with chrominances from similar images. For nighttime images that we do not have matching daytime chrominances, we discuss two techniques that use machine-learning to estimate chrominances from image priors that contain similar environments. For videos in non-visible spectra, we also discuss using machine vision to obtain the desired relative luminances of the visible spectrum.

By combining all these techniques in order, as shown in Figure 2, we can move towards making nighttime images resemble daytime images. First, dark footage can be noise reduced and then tone mapped to daytime luminance levels. Via colorization and further image enhancement, these videos can then take on more natural daytime image characteristics.

We now present the algorithmic details of each of our techniques followed by Section 3.6, where more detailed results of the techniques are shown.

3.1 Noise Reduction

Noise reduction has traditionally been in the domain of 2D image processing, whereby nearby pixels are used to both detect and then attenuate noise. However, noise reduction is possible, and in many cases preferable, if performed purely temporally. In the case of video from a stationary camera, each pixel is seen multiple times, allowing frame averaging to cumulatively reduce noise, a technique that has become feasible due to the low cost of memory. However, camera motion and scene motion cannot be strictly averaged out without introducing frame-to-frame ghosting artifacts. Therefore, we introduce noise filtering that adaptively handles motion with varying degrees of spatial filtering, and transitions to purely temporal filtering for static objects.
In this chapter, we also introduce a conceptual model called the Virtual Exposure Camera (VEC). Because our noise reduced video will be amplified via a tone-mapping algorithm in a later process, the noise reduction can be tailored to fit that amplification. Specifically, more aggressive noise reduction is performed in areas that will receive the most amplification, to remove quantization errors and uncover details. Combining pixels together to remove noise and

amplify intensity (extend exposure time) without introducing ghosting constitutes the central focus of the Virtual Exposure Camera.

The VEC system brings out hidden details that are barely noticeable in video frames due to underexposure and noise. It also synthesizes perceptually plausible and temporally consistent renditions of each video frame. Our method significantly enhances low dynamic range and noisy videos, making previously unwatchable material acceptable. The quantity of noise present affects the quality of the result, but so long as the noise is zero-mean, our method restores details obscured by noise and darkness.

Figure 3: A frame from a video processed using virtual exposures. From left to right: Original frame, histogram stretched version, pseudocolor version (red = number of temporal pixels integrated, green = number of spatial pixels integrated), and our result after our Virtual Exposure Camera processing.

3.1.1 Related Work

There is a long history of noise filtering methods throughout the signal processing literature. We are most interested in edge-preserving filters from the anisotropic diffusion and bilateral filter families. Anisotropic diffusion of images [Perona and Malik 90] provides an iterative filtering method that adapts to the image's gradient. Bilateral filtering [Tomasi and Manduchi 98] provides a single-step noise removal process that shares many visual and mathematical qualities with anisotropic diffusion [Barash 02]. However, both of these methods are designed for single images and not for videos. The Trilateral filter [Choudhury and Tumblin 03] builds on the bilateral filter model by biasing its kernel away from edges and dynamically choosing the kernel's size in an attempt to model signals as piecewise linear rather than piecewise constant functions. Other modifications have been proposed to improve the standard bilateral filter's ability to handle noise [Boomgaard and Weijer 02] [Francis and Jager 03]. We combine the attributes of median filters with the bilateral filter. A "bilateral median" filter was described by Francis and Jager [03], but it used a weighted median for summation purposes, unlike ours that uses it to establish a dissimilarity value. Spatio-Temporal Anisotropic Diffusion [Lee et al 98] used a three dimensional kernel to remove video noise, treating temporal and spatial dimensions similarly. Instead, we adapt from temporal to spatial filtering.

The idea of using multiple temporally adjacent frames to enhance knowledge about a pixel's true or desired value was considered in Cohen et al [03]. Multiple images were registered and then each pixel of the output image was computed as a function of its temporal neighbors. Sand and Teller [04] discussed a video matching method for aligning slightly different video sources. Specifically, it contains a robust system for frame-to-frame alignment. We handle moving cameras by warping spatio-temporal volumes as described by Bennett and McMillan [03].

Other video filtering approaches have appeared that use temporal filtering. Dubois and Sabri [84] performed nonlinear temporal noise filtering assisted by displacement estimation. Each pixel is combined temporally using a recursive low-pass temporal filter weighted by the reliability of the displacement estimate. This method requires well-exposed, easy-to-track video to correctly filter. Our method adapts from temporal to spatial filtering to be robust to tracking inaccuracies. Jostschulte et al [98] presented a spatio-temporal shot noise filter that first spatially and then temporally filters video while preserving edges that match a template set. A motion-sensing algorithm is used to vary the amount of temporal filtering. We prefer to only use temporal filtering when possible and adapt the mix of temporal and spatial filtering based on a tone-mapping objective and local motion characteristics.

Recently, Eisemann and Durand [04] and Petschnigg et al [04] have developed methods to remove noise and improve the dynamic range of underexposed images by incorporating features derived from properly exposed "flash images". The extent of noise removal depends on how well exposed a given region is in the flash image. Furthermore, the underlying luminance model used in the processing is not HDR, either explicitly (as in previous tone-mapping systems) or implicitly (as in our case). It is also unclear how to extend these methods to video sequences. The goal of our virtual exposure approach is similar to these methods, but we incorporate temporal information instead of flash image features to improve the exposure. Thus, the illuminations of our enhancements are consistent with the original source.

3.1.2 The Virtual Exposure Camera Model

The Virtual Exposure Camera (VEC) is our conceptual model for analyzing and enhancing low dynamic range (LDR) video. Many common applications result in LDR videos. For instance, filming under uneven lighting is difficult because certain objects are often poorly exposed compared to the well-lit objects. LDR video also results from high speed imaging, where fast shutter speeds are desirable. Small aperture video, to increase depth-of-field, can also lead to LDR video. Insufficient lighting scenarios, common in surveillance applications, also lead to underexposed videos.

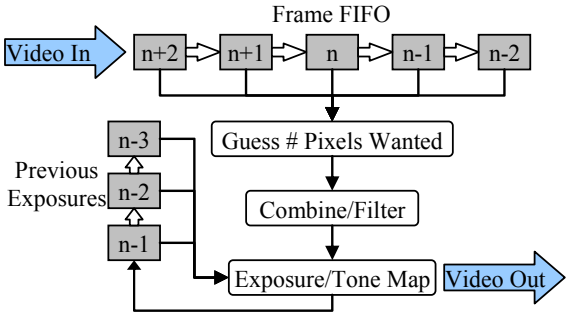


Figure 4: The VEC model for processing LDR video. Since no single frame contains sufficient information for noise reduction and tone mapping, processing is done with knowledge of recent frames and how tone mapping was applied. Rudimentary tone mapping is performed before filtering to guide the adaptive filter's settings.

3.1.2.1 LDR Video Noise Characteristics

The LDR videos we are interested in processing have a small signal-to-noise ratio and low precision. Our system also enhances videos with "peaky" histograms. Such scenes are composed of elements that span a significant dynamic range, but the combination of exposure settings and quantization leads to low precision renditions of all elements.

There are a variety of noise sources in CCD and CMOS sensors that confound imaging in low-light situations, such as readout, photon shot, dark current, and fixed pattern noise in addition to photon response non-uniformities [Reibel et al 03]. We assume that dark current noise and fixed pattern noise can be removed via subtraction of a reference dark image at the same temperature and exposure setting. Photon shot and readout noise are our primary problems, but we assume they are zero-mean, so if we can get multiple samples of the same pixel from temporally adjacent frames, we can average out the error. A significant problem for dealing with dark areas captured with CCDs is that the amplitude of sensor read noise is independent of exposure whereas photon shot noise varies linearly with exposure time. Read noise is more significant than shot noise at very dark pixels. Thus, for the darkest pixels, the SNR is comparatively smaller.

Computing the mean of n samples will improve the precision of the luminance readings by a $\sqrt{n}$ factor. These assumptions are not true for compressed video footage, where quantization is non-uniform across frequencies. We assume a linear camera response, which is true for raw CCD samples, but not for the hidden post-processing found in many camcorders.

3.1.2.2 Synthesizing Virtual Exposures

Our VEC model identifies poorly exposed regions of video and increases precision by simulating longer exposure times. This simulation involves temporal integration of the contributions of as many pixel values as would have been sampled over the interval of the longer exposure.

We process a spatio-temporal volume implemented as a FIFO queue (Figure 4), where filtering occurs in the current frame but with knowledge of the frames that come before or after it (in a real-time, low latency system, the future might not be known). Therefore, the processing of a pixel can benefit from information in adjacent frames while also ensuring that tone mapping is temporally consistent. Pixels are indexed using (x,y,t) notation, with t being the frame number.

When integrating the contributions of multiple pixels together to simulate a longer exposure time, pixels that come before and after temporally can often be used. However, since some frames capture individual pixels with varying noise contributions, it is advantageous to exclude the noisiest pixels from the integration. Similarly, pixels that change due to object motion should not be included, to avoid blurring and "ghosting" artifacts.

Given LDR video, we apply a tone mapping algorithm targeted at improving poorly exposed areas and handling noise. Such a tone mapper is discussed in Chapter III.

Prior to filtering, we estimate a gain factor for each pixel that scales its original luminance to achieve the pixel's final filtered output level. Because we cannot know this before filtering, we choose to estimate the filtered and tone mapped luminance by applying a spatially uniform tone

mapping function m(x,ψ) to a Gaussian blurred version of the image. This gain factor is used by our non-linear filter, described in Section 3.1.3, to determine how many pixels are additively combined thus establishing a per-pixel exposure time. We call this gain value λ and we use it to establish our adaptive filter's support.

3.1.3 The ASTA Filter

Our virtual exposure filter seeks out similar pixels to integrate. Two major factors affect how ASTA filters: how many pixels it wants to combine and if these pixels are in an area of the image with motion. ASTA adapts by transitioning between temporal-only and spatial-only bilateral-inspired filtering while choosing parameters based on local illumination.

3.1.3.1 The Spatial Bilateral Filter

ASTA is based on the edge-preserving bilateral filter [Tomasi and Manduchi 98]. The bilateral filter maintains edges by performing a Gaussian convolution but weights the contributions of pixels by how different their intensities are from the intensity at the center of the kernel. Although a simple subtractive difference is often used to measure this difference of intensities, we generalize this notion to include non-photometric differences which we treat as dissimilarity values. A dissimilarity value is any relationship that satisfies the following properties: D(x,x) = 0 and D(x,y) = D(y,x). A dissimilarity is metric if the triangle inequality holds: D(x,y) + D(y,z) ≥ D(x,z).

The spatial bilateral filter (for a pixel s), with a subtractive dissimilarity value D(p,s), is shown in Equations 1 and 2:

$$B(s,\sigma_h,\sigma_i)=\frac{\sum_{p\in N_s} g(\|p-s\|,\sigma_h)\,g(D(p,s),\sigma_i)\,I_p}{\sum_{p\in N_s} g(\|p-s\|,\sigma_h)\,g(D(p,s),\sigma_i)} \qquad (1)$$

$$g(x,\sigma)=e^{\frac{-x^2}{2\sigma^2}}/\left(\sigma\sqrt{2\pi}\right)$$

$$N_s=Kernel=\begin{bmatrix} p_x=[s_x-k,s_x+k] \\ p_y=[s_y-k,s_y+k] \end{bmatrix}$$

where $$D(p,s)\equiv I_p-I_s \qquad (2)$$

Three variables control the bilateral filter's operation. First, σ_h controls how quickly the spatial Gaussian falls off. The second, σ_i, controls the Gaussian dissimilarity weighting. It attenuates the contributions of neighboring pixels that are that are too different and is typically chosen based on an estimate of the signal's SNR. Finally, k determines kernel size.

The bilateral filter does a good job of removing noise while preserving edges, but it is incapable of removing shot noise from a signal (Figure 5). When the filter kernel is centered on an outlier pixel, the intensity Gaussian will exclude all other values, leaving it unchanged, which accentuates it compared to the otherwise cleaned signal.

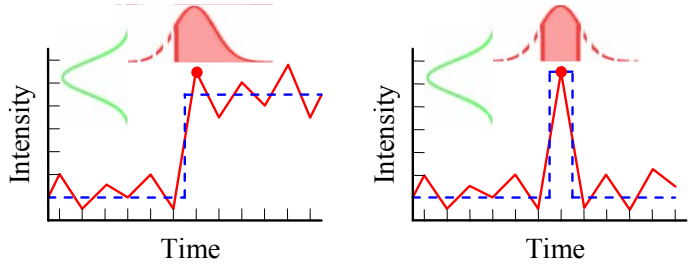


Figure 5: Left: The bilateral filter recovers the signal (blue) from the noisy input (red). Right: The bilateral filter is unable to attenuate the shot noise because no other pixels fall within the intensity dissimilarity Gaussian.

3.1.3.2 Bilateral Filtering in Time

In the case of a fixed camera, the best estimate of a pixel's true value is predicted from those pixels at the same location in different frames. In the absence of motion, a simple average of all pixels at each (x,y) coordinate through time gives an optimal answer, assuming zero-mean noise. However, averaging in the presence of motion creates "ghosting" artifacts. Our solution is to consider changes in a pixel's value due to motion as "temporal edges". A bilateral filter maintains edges while providing noise reduction in areas with small amplitude noise. Thus, we employ a temporal 1D-bilateral filter as a primary component in our noise reduction process.

A difficulty of applying a temporal bilateral filter is choosing an appropriate value for σ_i (the dissimilarity falloff) that simultaneously removes noise while preserving motion based entirely on differences of pixel luminance. If σ_i is too large, "ghosting" still results, and if σ_i is too small, noise will remain. Such a simple σ_i does often not exist for noisy video.
An alternative is to filter video with a volumetric bilateral kernel that operates in spatio-temporal volumes, much like how anisotropic diffusion was extended to 3D by Lee et al [98]. However, this symmetric approach does not take into account the difference in sampling density between space and time in a spatio-temporal volume.

3.1.3.3 Alternate Dissimilarity Values

As a solution to the typical bilateral filter's inability to remove shot noise, we introduce an alternate dissimilarity value D(p,s) in the bilateral filter. Instead of using the simple intensity difference, we substitute an arbitrary function that returns a value for each pair of pixels in a video or image that may or may not be solely intensity-based.

For example, the dissimilarity value could be the difference between p and some statistic of the local spatial neighborhood around s, making the filter more robust to shot noise. We use a median-centered bilateral filter that uses a small kernel median filter centered at s to improve quality in noisy image areas. The problem of choosing the intensity at the bilateral filter's center as the sole reference was discussed by Boomgaard and Weijer [02], but no suggestion of an alternative statistic was given. A wide variety of statistics could be applied to choose the s pixel's intensity, such as local minima, local maxima, or even other bilateral filters. Even measures not directly associated with luminance could be used.

3.1.3.4 Spatial Neighborhood Dissimilarity Value

We use a different dissimilarity value in our temporal bilateral filter. Specifically, the method is to compare the local spatial neighborhoods centered at the same pixel in different frames. Equation 3 shows our normalized Gaussian weighted dissimilarity for an n × n neighborhood and a temporal edge tolerance of σ_e.

$$D(p_{xyt}, s_{xyt}) = \frac{\sum_{x=sx-n}^{sx+n} \sum_{y=sy-n}^{sy+n} g\left(\left\|x - p_x, y - p_y\right\|, \sigma_e\right) \left|I_{x,y,pt} - I_{x,y,st}\right|}{\sum_{x=sx-n}^{sx+n} \sum_{y=sy-n}^{sy+n} g\left(\left\|x - p_x, y - p_y\right\|, \sigma_e\right)} \quad (3)$$

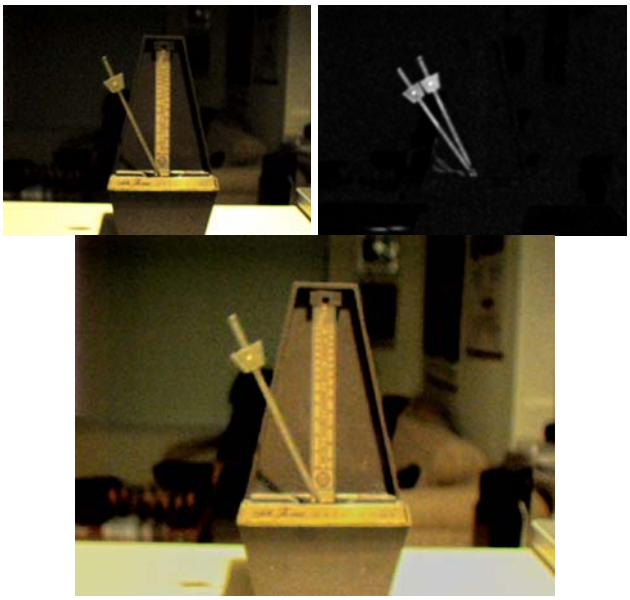


Figure 6: Illustration of our spatial neighborhood dissimilarity value used in temporal filtering. The original frame is shown in the upper left. Each (x,y) for a pair of nearby frames are shown in the upper right. Two metronome arms are seen because the dissimilarity value is based on absolute value. The bottom image is the same frame processed using ASTA and our tone mapper.

The difference between two pixels' intensities does not provide enough information to judge if they are significantly different. However, by comparing spatial neighborhoods, a judgment can be reached. Thus if only a small percentage of pixels change, we assume it to be noise and integrate into the filter. If many pixels change, we assume it to be a more significant event, and no blending occurs. For clarification, despite the fact we are comparing neighborhoods, it is only the pixels at the center of each neighborhood that will ultimately be blended together. The neighborhood size, often between 3 and 5, can be varied depending on noise characteristics, as can σ_e (usually between 2 and 6). Our dissimilarity value is inspired by correspondence measures frequently used in stereo imaging. We have used Sum of Absolute Differences (SAD) and Sum of Squared Differences (SSD). We implemented both versions and got similar results, although SSD occasionally created artificially sharp edges. Figure 6 illustrates our SAD version.

3.1.3.5 Implementing ASTA

The VEC model determines how many pixels should be combined to achieve our tone map brightness target. If only temporal bilateral filtering with the spatial neighborhood dissimilarity value is used, and it is in an area of high motion, only the center pixel of the kernel will make a sizable contribution to the result. In this case, it would not integrate enough pixels to achieve the desired gain factor. To overcome this problem we instead use an Adaptive Spatio-Temporal Accumulation filter (ASTA) that adapts to its surroundings to find enough pixels in the presence of motion. For a static pixel, it reduces to a temporal bilateral filter with the spatial neighborhood difference dissimilarity value. However, if it does not find enough similar pixels to achieve the desired exposure based on the size of the normalizing factor in the denominator of 1, it transitions to a spatial-only median-centered bilateral filter, as shown in Figure 7. Like Yee et al [01], we also exploit the psychophysical phenomenon that in areas of motion, the human visual system's ability to perceive high frequencies is reduced. Thus, in areas with insufficient temporal information due to motion, we can transition to spatial filtering.

One way to conceptualize ASTA is as a voting scheme, where each vote is a measure of the support of the filter. Before ASTA is run on a pixel, we determine how many votes (pixels) are required (defined as λ, Section 3.1.2.2). The temporal bilateral filter gathers some votes, and if they are not sufficient, more votes are gathered from the spatial bilateral filter.

The number of votes desired is defined as $\lambda \times g(0,\sigma_h) \times g(0,\sigma_i)$. The factor $g(0,\sigma_h) \times g(0,\sigma_i)$ is our definition of a vote because it is the contribution to the denominator of the bilateral filter from a pixel that is an exact match in space and intensity (D(x,y)=0). The larger the dissimilarity value, the lower its contribution to the denominator is. Thus, by analyzing the denominator of a bilateral filter, we can determine if a sufficient number of votes were tallied. ASTA is thus formalized in Equation 4. The terms n and d represent the numerator and denominator of Equation 1, respectively.

$$\frac{n_T}{d_T} = temporalBilateral(x, y, t, \sigma_h, \sigma_i)$$

$$\frac{n_S}{d_S} = spatialBilateral(x, y, t, \sigma_h', \sigma_i')$$

$$\omega = \lambda \times g(0, \sigma_h) \times g(0, \sigma_i)$$

$$ASTA(x, y, t, \lambda, \sigma_i, \sigma_i') = \begin{cases} \dfrac{n_T}{d_T}, & d_T \geq \omega \\ \dfrac{n_T + n_S}{d_T + d_S}, & d_T < \omega \; AND \; d_T + d_S < \omega \\ \dfrac{n_t + n_s \dfrac{(w - d_t)}{d_s}}{w}, & d_T < \omega \; AND \; d_T + d_S \geq \omega \end{cases} \quad (4)$$

ASTA changes its filtering settings based on the number of pixels it wants to combine. First, not every pixel could ever get a full vote, because even though it may have the same neighborhood it is attenuated by the distance Gaussian. Therefore, we choose the temporal filter kernel size and Gaussian σ_h dynamically such that if every comparison were a perfect match, $Dt \approx 2 \times \omega$. Similarly, if the vote count for the temporal bilateral comes up short, the spatial bilateral

attempts to have the remaining number of votes fall within the area of one standard deviation of its distance Gaussian by dynamically choosing σ_h'. The remaining sigmas, σ_i for the temporal bilateral (and σ_e for its dissimilarity value) and σ_i' for the spatial bilateral, are held constant in each video's processing.

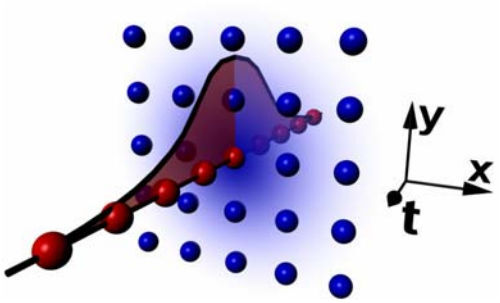


Figure 7: Illustration of the temporal-only and spatial-only nature of ASTA. The temporally filtered red pixels are preferred to be integrated into the filter, but if not enough are similar to the center of the kernel, the blue spatial pixels begin to be integrated.

Temporal bilateral filters are run on the image's luminance and mapped to each channel, but only spatial filtering is done on each color channel. Furthermore, spatial filtering is done in the log domain, whereas temporal filtering is not.

So far, we have assumed that the camera used to capture footage is stationary, assuring spatial correspondences for background pixels. For moving cameras, feature tracking is used. Sand and Teller [03] detail a system for finding accurate frame-to-frame correspondences which can identify temporal neighbors. In our system video registration and alignment takes place prior to noise reduction. We only consider "high-confidence" trackable points (as determined by OpenCV's GoodFeaturesToTrack()). We then select high-confidence optical-flow vectors (as determined using OpenCV's feature tracking) that correspond to the trackable points that occur on the dominant flow field (typically the background). Finally, we select the mean of this feature set as a translation for each frame. Our approach removes only the dominant motion, although more complex tracking methods could also be used. We used the spatio-temporal video editing system of [Bennett and McMillan 03] to do this. Once stabilized, video can be processed and then the stabilization can be removed. Any residual motions or misalignment are treated as moving objects by our ASTA filter.

3.1.4 Implementing ASTA in Software

In this section, we present the software structure by which ASTA is implemented. Design choices were made for the utmost accuracy of the results and not for overall speed. Alternatively, the new fASTA system implementation (Section 6.2), currently in development, does not solve the problem in nearly as structured a manner, but it runs faster. Because fASTA's inner workings are still in flux and because its execution order is considerably more convoluted, the original design is presented here for clarity.

The main data structure underlying ASTA is the spatio-temporal volume, where each video frame constitutes a single planar slice of a 3D volume. The advantages of working in this space are that time and space can be handled in a consistent manner and that, via Bennett and McMillan [03], these volumes can be warped to achieve local registration. The warping process for non-stationary cameras is discussed in more detail later in this section.

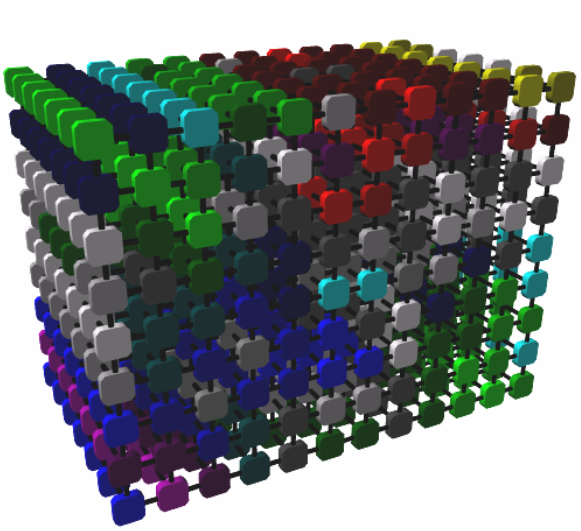


Figure 8: Visualization of the spatio-temporal data structure used in ASTA processing. Temporally and spatially adjacent pixels can be read and written with the same constant access time, allowing for efficient adaptive processing.

However, with the advantages of spatio-temporal volumes come some disadvantages as well. Primarily, the memory consumption of a multi-second, high resolution, high bit-depth video can easily reach into the gigabytes. This is because, in order to have random access to any pixel with a fixed access time, all frames must be kept in an uncompressed state. In implementation, to allow videos of arbitrary length to be opened and processed, a per-frame caching scheme is used. When a file is initially opened, the software creates a separate image of each frame on disk. When any pixel is accessed, it is verified to be in core RAM. If it is not, it is loaded from disk and the oldest frame in memory is written back out to disk (i.e. a FIFO cache). For this reason, the filters attempt to never have a kernel size greater than the cache size, else severe disk thrashing will result from loading and unloading entire frames.

The structure of actual execution is as described in Section 3.1.3.5. The pseudo-code below outlines the general structure of the pipeline of the application. However, for the sake of clean pseudo-code, only the luminance channel processing is shown. In reality, for RGB video, the Spatial Neighborhood Dissimilarity is done only in the luminance domain, which is less noisy than any of the individual channels. The weightings determined by that dissimilarity are used uniformly for each of the color channels. The Median Centered Bilateral can be run on either the luminance (and then propagated to each of the colors as before) or it can be run separately on each channel. This decision should be made based on the chrominance quality of the sensors.

Specifically, lower-end sensors typically have poorer red and blue response than green due to Bayer patterning, whereas high end 3CCD color sensors have better color accuracy.

ASTA Psuedo-Code Implementation

```
gaussian(x,sigma)
       return((e^-((-x^2)/(2*sigma^2))/(sigma*sqrt(PI));

spatialNeighborhoodDissimilarity(x,y,z1,z2)
   sum = 0;
   totalMagnitude = 0;
   for each u = x-3 to x+3
               for each v = y-3 to y+3
                      magnitude = gaussian(sqrt((u-x)^2+(v-y)^2), SND_SIGMA);
                      totalMagnitude = totalMagnitude + magnitude;
                      sum = sum + magnitude * (luminance[u,v,z1]-luminance[u,v,z2])^2;
   return (sum / totalMagnitude);

ASTA(x,y,z)
       targetLuminance = simpleTonemap(luminance[x,y,z])
       gainFactor = targetLuminance / luminance[x,y,z];
       singleVote = gaussian(0,ASTA_TEMPORAL_DIFF_SIGMA) *
                        gaussian(0,ASTA_TEMPORAL_FALLOFF_SIGMA)

       n_temporal = 0;
   d_temporal = 0;
   temporalWindowSize = (gainFactor * KERNEL_OVERESTIMATION) / 2
   for each w = z-temporalWindowSize to z+temporalWindowSize
      temp = gaussian(w-z, gainFactor) *
         gaussian(spatialNeighborhoodDissimilarity(x,y,z,w),ASTA_TEMPORAL_DIFF_SIGMA));
               d_temporal = d_temporal + temp;
               n_temporal = n_temporal + temp * luminance[x,y,w];

      if d_temporal >= gainFactor*singleVote
                      return (n_temporal / d_temporal);
      else
                      remainingVotes = (gainFactor*singleVote-d_temporal) / singleVote;
         n_spatial = 0;
         d_spatial = 0;
         spatialWindowSize = sqrt(remainingVotes/PI);

         for each u = x-spatialWindowSize to x+spatialWindowSize
            for each v = y – spatialWindowSize to y+spatialWindowSize
               medianLum = median(all luminances within 3x3 neighborhood of (x,y))
               temp = gaussian(sqrt((x-u)^2+(y-v)^2), remainingVotes / 2) *
                      gaussian(luminance[u,v,z] – medianLum, ASTA_SPATIAL_FALLOFF_SIGMA);
                                    d_spatial = d_spatial + temp;
                                    n_spatial = n_spatial + temp * luminance[u,v,z];

            if(d_temporal+d_spatial) >= gainFactor*singleVote
                                    return (k_temporal+k_spatial) / (j_temporal + j_spatial;
                              else
                                    return (k_temporal + k_spatial*(gainFactor*singleVote –
                                   d_temporal) / d_spatial) / (gainFactor * singleVote);
```

Table 1: Typical ASTA Constant Values

ASTA_TEMPORAL_DIFF_SIGMA	4.0
KERNEL_OVERESTIMATION	2.0
ASTA_SPATIAL_FALLOFF_SIGMA	3.0

Non-stationary cameras are currently handled by warping the video so that the primary flow field (typically the background) remains fixed from frame-to-frame. The Proscenium system [Bennett and McMillan 03] can perform this warping when assisted by human-in-the-loop. Proscenium pre-processes video to typically find the 500 most trackable points in a video (as defined by OpenCV's GoodFeaturesToTrack()). The user can select a set of these points to fully define the registration homography (up to a least-squares fit of a projective transform). OpenCV's Lucas-Kanade tracker handles the actual tracking, and the least-squares fit is thus solved and applied to spatially warp each plane of the spatio-temporal volume. This transform is also stored for later use. Once the new spatio-temporal volume has been determined, the same ASTA processing can be applied as was used on the stationary camera video. However, once the processing is complete, the stored homographies are inverted and applied to the spatio-temporal volume. This returns the original camera motion. For specifics on how best to handle the sampling during these spatio-temporal warping steps, please refer to Bennett and McMillan [03].

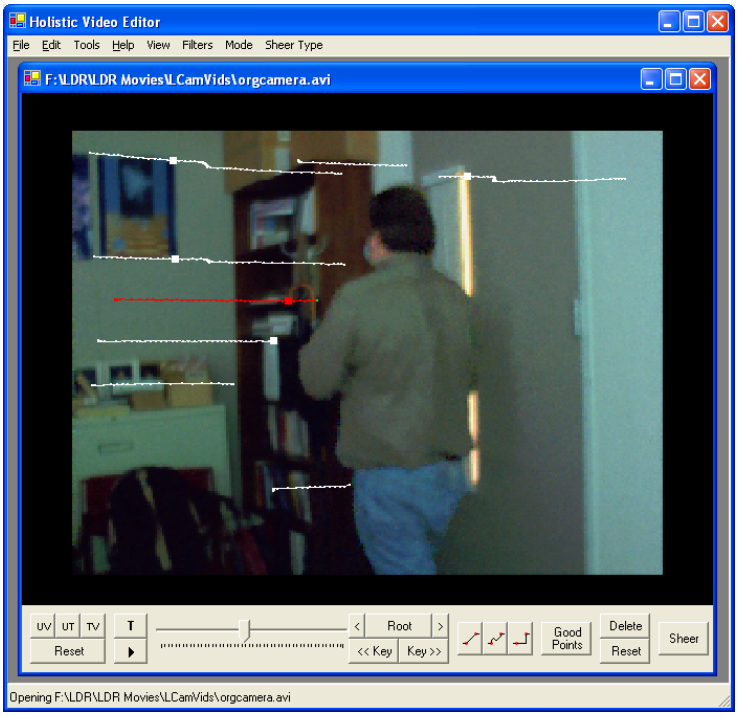


Figure 9: Screenshot of point tracking within the Proscenium framework. Each of the white and red lines represents a valid track through time. These tracks will be used to stabilize the primary flow field, which in this case is the background.

3.2 Tone Mapping

High dynamic range (HDR) imaging, processing, and display have recently received considerable attention. An implicit assumption of most HDR systems is a sizeable signal-to-noise ratio achieved via long exposures in low-light areas. Generally, multiple low dynamic range (LDR) images with different exposure settings are combined to generate a single HDR image, which implies a static scene. Multiple video frames captured at the exact same exposure can also contain a large range of luminances, but most of the details are difficult to see because they are comparatively too dark. We address the problem of enhancing such videos. Aside from the noise characteristics of dark videos, there is a surprising commonality between HDR and LDR imaging. In this chapter, we develop methods for enhancing LDR video to simulate the characteristics of individually tone-mapped HDR video frames, for applications in surveillance, filmmaking, forensics, and high-speed imaging.

Humans simultaneously perceive regions with high luminous intensities alongside intensities several orders of magnitude lower by spatially adapting the visual field's local sensitivity. Modern digital still cameras, however, rely on a single exposure time across the entire frame and photosites with uniform sensitivities. This necessitates that multiple images be taken at varying exposures to capture the full nuance of HDR scenes. This is problematic for dynamic scenes, where it is seldom possible to capture multiple exposures. Furthermore, HDR construction assumes an abundance of light and/or exposure intervals long enough to cancel the random noise fluctuations characteristic of image sensors. Once acquired, the problem becomes the accurate depiction of HDR results on LDR displays through tone mapping.

Alternatively, HDR images can be assembled by additively combining multiple uniformly exposed digital images [Liu et al 03] [Jostschulte et al 98]. This approach is particularly compatible with digital video, but it has been largely overlooked since it requires processing O(N) source images compared to the O(log(N)) of variable exposure methods. Using the techniques described in Chapter II, it is possible to combine the contributions of multiple frames in order to simulate these longer exposure times. However, the number of frames to combine needs to be known so that the proper gain ratio is applied.

In this chapter, we describe a method for finding a per-pixel amplification term that targets maximizing the visible contrast in an LDR video. This term can be used to adaptively drive ASTA noise reduction or to directly enhance less noisy LDR videos. This is performed in a temporally coherent way that specifically avoids amplifying what could be residual high-frequency noise while preserving image structure.

3.2.1 Related Work

HDR representations have long been recognized as essential for accurately modeling light transport [Ward 91]. More recently, Debevec and Malik [97] developed accurate methods for assembling HDR images from a series of still photographs with increasingly long exposure times.

The problem of mapping an HDR image for display on devices with limited dynamic range was formalized by Tumblin and Rushmier [93], and has led to a variety of spatially uniform [Drago et al 03] and spatially varying [Tumblin and Turk 99] [Durand and Dorsey 02] [Fattal et al 02] tone mapping approaches. A variety of methods have been proposed to tone map HDR images so that the maximum amount of information is visible on a monitor. Retinex theory, such as in the multiscale Retinex [Jobson et al 97], suggests that a Gaussian-like kernel can be convolved at each point in the image and subtracted from the original image in log space, providing for a more "viewable" version of a still image. The advantage of the Retinex approach is that it is non-iterative, but it can also generate unwanted edge blurring artifacts. Durand and Dorsey [02] built a similar system, but used an edge preserving bilateral filter to maintain sharp edges. Pattanaik et al [00] presented an approach that mimics the time dependent local adaptation of the human visual system. They also discussed temporal coherence to avoid introducing frame-by-frame tone mapping "flicker". In gradient domain HDR compression [Fattal et al 02], the gradient of an image is attenuated and then reintegrated. They also described a modification for improving images that already use a display's full dynamic range. Raskar et al [04] also used gradient domain methods, but to fuse day and night images together— adding daytime context to nighttime footage.

Researchers have also constructed actual high dynamic range video capture systems. Kang et al [03] built a system based on a camera that could sequence through different exposure settings. Once the images were registered using optical flow, it was possible to combine exposures to improve the dynamic range. The small number of frames combined suggests that a high signal-to-noise ratio (SNR) was assumed, and therefore, it would only be useful for well-lit scenes. Nayar and Branzoi [03] presented a system whereby a computer controlled LCD panel was placed in front of the CCD. The per-pixel transparency was varied to modulate the exposure of image regions based on the previous frame's luminance. They also discussed a local and global tone mapping approach that addresses temporal coherence issues. Using LCDs implies attenuation of the incoming light, thus further complicating low-light imaging. Nayar and Branzoi [04] later suggested a second variant using a DLP micromirror array to modulate the exposure, via time-division multiplexing (like a camera shutter), throughout the image. In theory, such systems could provide continuous exposure control at each pixel compared to our discretized exposure settings. However, they require additional hardware and are strictly causal; whereas our virtual exposure approach allows the incorporation of future information into virtual exposure decisions, assuming a constant latency.

Acosta-Serafini et al [04] described an HDR camera that selectively resets a pixel based on a prediction of when it will saturate. The reset interval and the digitized pixel level combine to form a floating-point value. They primarily focused on high-speed, HDR sensing and do not specifically address low-light situations. Liu et al [03] combined high-speed samples to reduce

noise and improve dynamic range. Their approach is similar, but much lower-level than ours. It depends on specific imaging device features such as high-speed non-destructive reads. It also relies mostly on linear filters, and uses only single pixel areas to detect motion. In contrast, our method uses bilateral filtering, considers a larger context for motion detection, and targets a tone-mapped objective. Bidermann et al [03] described an HDR high-speed CMOS imager platform with per-pixel ADCs and storage, which could use the Liu et al [03] algorithm and targets well-lit scenes.

3.2.2 LDR Tone Mapping

Our tone mapping approach considers that SNR varies with intensity. Thus, details in dark regions are less accurate than those in brighter regions. A tone mapper specialized for underexposed video should therefore associate a confidence level for details based on their luminous intensity. For instance, in the brightest areas of a video where the CCD received a reasonable exposure, the mix of details and large-scale features should be adjusted to achieve the tone mapping objectives. In darker areas the details should be attenuated to suppress noise.

Using the tone-mapping approach of Durand and Dorsey [02], it is possible to separate an image into details and large scale features. Subtracting the original log-image from a bilaterally filtered log-image provides an estimate of the image details. Durand and Dorsey then attenuate the large scale features by a uniform scale factor in the log domain to reduce the overall contrast of the HDR image, but leave the details untouched. This is not a problem for low-noise source images. In contrast, our LDR tone-mapping processes the details and large scale parts with different pipelines that attenuate details based on their estimated accuracy, as determined by local luminance, and it attenuates the large scale features to achieve the desired contrast. These two signals are then remixed to form the final output.

The same nonlinear mapping function, with independent parameters, is used to attenuate image details and to adjust the contrast of large scale features. It obeys the Weber-Fechner law of just-noticeable difference response in human perception but provides a parameter to adapt the logarithmic mapping in a way similar to the logmap function of Drago et al [03] and Stockham [72]. The mapping is given by:

$$m(x,\psi) = \frac{\log\left(\frac{x}{xMax}(\psi - 1) + 1\right)}{\log(\psi)} \qquad (5)$$

The white level of the input luminance is set by xMax and ψ controls the attenuation profile. As shown in Figure 10, the shape of our detail attenuation and contrast mapping function, m(x, ψ), is similar to a traditional gamma function, but it exhibits better behavior near the origin. As noted by Drago [03] the high slope of standard gamma correction for low intensities can result in loss of detail in shadow regions. This is particularly troublesome for underexposed images like those we target.

Tone mapping (Figure 11) begins by extracting the luminance of each frame and the chrominance ratio of each color component as discussed by Eisemann and Durand [04]. A

bilateral filter is then applied to the log-image to extract the large scale image features. A temporal bilateral filter, with narrow support (small σ_i), is then applied to maintain temporal coherence. This result is then subtracted from the log luminance of the original image to yield the detail features.

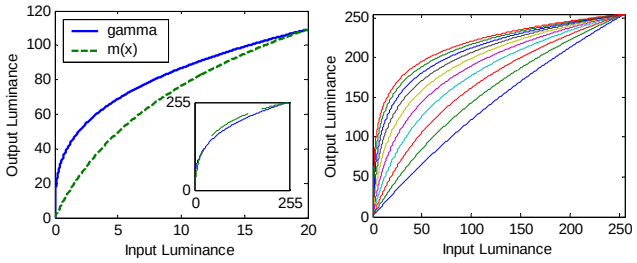


Figure 10: Plots showing our nonlinear mapping function. The left plot shows how our function does not have as severe a slope for luminances near 0 as does gamma correction as to not over accentuate dark regions (γ=2.0 for gamma correction, ψ=64 for $m(x, \psi)$). The inset shows that over the rest of 0-255, they are mostly similar. The right plot shows a family of $m(x, \psi)$ curves of ψ=2 (the most linear) through ψ=1024 (the most curved).

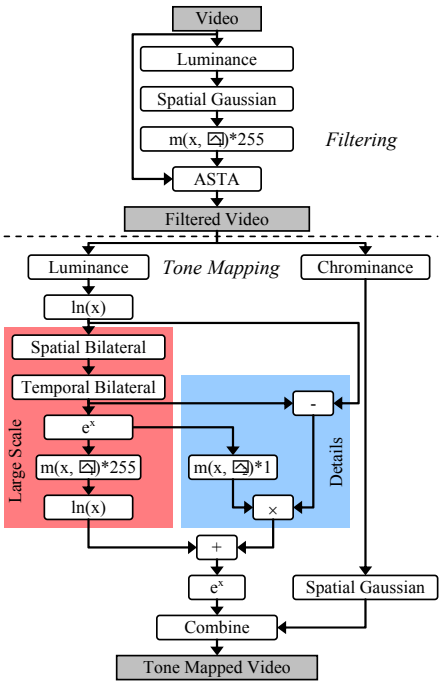


Figure 11: A flowchart of the entire process for creating virtual exposures, including detail of the LDR tone-mapping process. The highlighted areas show the different processing paths of large scale and detail features.

The linear intensities of the large scale features are next uniformly tone mapped using Equation 5, with a ψ_1 of approximately 40. The log-intensities of the details are attenuated based on the brightness of the linear large scale features. With a linear attenuation, a pixel with a brightness of .5×maximum would have half its high frequency masked. Since confidence in detail degrades at dark values, we attenuate with the curve in Equation 5 with a different ψ_2 (about 700.0), resulting in a steep roll off for low intensities.

The log large scale features and log detail features are recombined to generate the final output luminance. Noise in the chrominance is attenuated via standard Gaussian blurring. Finally, the luminance and chrominance ratios are then recombined.

3.2.3 Implementing Tone-Mapping in Software

Our tone mapping can be used in one of two ways. First, the techniques presented here can be used on any dark video or image. If noise is present (which is almost certain for dark videos) it should be run in conjunction with the ASTA algorithm as shown in Figure 12. In this configuration, the tone mapping is actually run twice. The first time is to estimate the gain at each pixel, which in turn determines the strength of the adaptive filter. It is run again on the noise-reduced video at the end of the pipeline again to increase the contrast of the output video without corrupt noise to influence fine texture features.

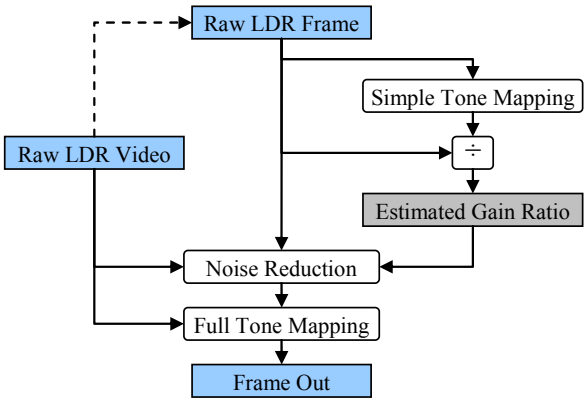


Figure 12: Pipeline of how tone mapping is commonly performed twice in the Virtual Exposure Camera infrastructure. The first time is to estimate the rigorousness of the algorithm and the second time is to do a cleaner final pass.

In implementing tone mapping, ASTA's spatio-temporal data structures are used as well as its disk caching scheme for supporting an unbounded numbers of frames (Section 3.1.4). The pseudo-code below implements the pipeline shown in Figure 13.

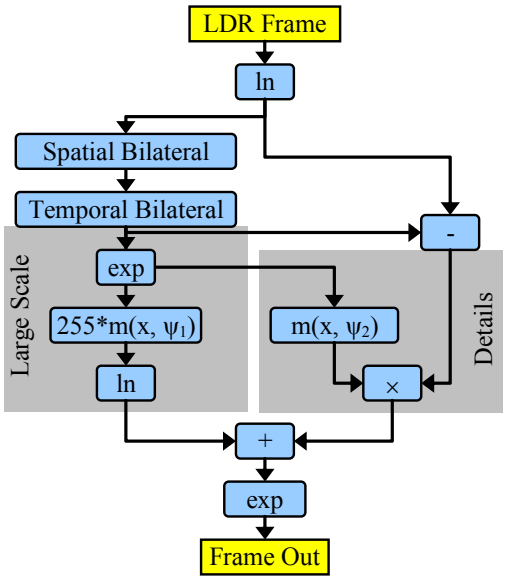


Figure 13: Processing pipeline for the full tone-mapping algorithm.

Tone Mapping Psuedo-Code Implementation

```
logSpatialBilateral(x, y, z, spatialSigma, intensitySigma)
   n = 0;
   d = 0;
   for each u = x–kernelSize to x+kernelSize
               for each v = y-kernelSize to y+kernelSize
                       temp = gaussian(sqrt((u-x)^2+(v-y)^2), spatialSigma) *
                                       gaussian(ln(luminance[u,v,z])-ln(luminance[x,
                                       y, z]),intensitySigma);
                       d = d + temp;
                       n = n + temp * ln(luminance[u,v,z]);
   return n / d;

temporalBilateral(x, y, z, temporalSigma, intensitySigma, input)
   n = 0;
   d = 0;
   for each w = z–kernelSize to z+kernelSize
                       temp = gaussian(w-z, temporalSigma) *
                                       gaussian(input[x,y,w]-input[x, y, z],intensitySigma);
                       d = d + temp;
                       n = n + temp * input[u,v,z];
   return n / d;

logmap_m(x, psi)
       // Assume we know xMax, the largest luminance in the image from which x originated
   return log((x/xMax)*(psi-1)+1) / log(psi);

fullToneMap(x,y,z)
       largeScaleVal=temporalBilateral(x,y,z, TONEMAP_TEMPORAL_SIGMA,TONEMAP_INTENSITY_SIGMA,
               logSpatialBilateral(x,y,z,TONEMAP_SPATIAL_SIGMA, TONEMAP_INTENSITY_SIGMA));
               // There is a type mismatch here.  In order to run the temporalBilateral, the
               // logSpatialBilateral must be known for all temporally aligned pixels.  Assume
               // for the sake of this pseudo-code that this has been solved and placed into
               // a cache.
       detailVal = ln(luminance(x,y,z)) – largeScaleVal;
       largeScaleVal = exp(largeScaleVal);
       detailVal = detailVal * logmap_m(largeScaleVal, DETAIL_PSI);
       largeScaleVal = ln(255*(logmap_m(largeScaleVal, LARGE_SCALE_PSI)));
       result = exp(largeScaleVal + detailVal);
               // If desired, perform a 1 pixel falloff Gaussian blur on the chrominance to
               // help remedy color sensor resolution issues
       return result;
```

Table 2: Typical Tone-Mapping Constant Values

TONEMAP_SPATIAL_SIGMA	3.0
TONEMAP_INTENSITY_SIGMA	0.5
LARGE_SCALE_PSI	400.0
DETAIL_PSI	700.0

For the initial step of ASTA tone mapping, where a tone mapping operator is needed to estimate the overall gain factor, a less complex process should be used. This is because the gain factor that determines the necessary noise reduction should be based on low frequency luminance, not high frequency luminance. For example, a bright pixel in a dark area still needs to be heavily filtered to verify it is not shot noise. Similarly, a dark pixel in a bright area does not need to be as filtered, as it more likely is texture. So, the process can be simplified to discard detail features and to use Gaussian smoothing as opposed to edge-preserving smoothing. The next pseudo-code shows this process as depicted in Figure 14.

```
logSpatialGaussian(x, y, z, spatialSigma)
   n = 0;
   d = 0;
   for each u = x–kernelSize to x+kernelSize
               for each v = y-kernelSize to y+kernelSize
                      temp = gaussian(sqrt((u-x)^2+(v-y)^2), spatialSigma);
                      d = d + temp;
                      n = n + temp * ln(luminance[u,v,z]);
   return n / d;

simpleLogmap_m(x,y,z)
       val = logSpatialGaussian(x,y,z, TONEMAP_SPATIAL_SIGMA);
       val = 255*(logmap_m(exp(val), LARGE_SCALE_PSI);
       return val;
```

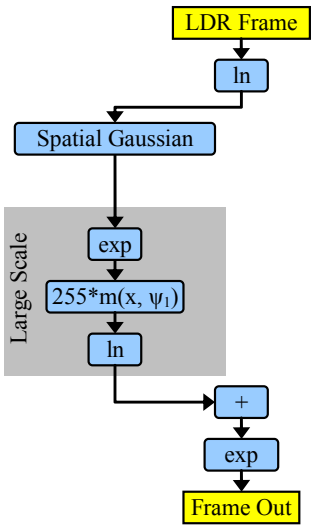


Figure 14: Processing pipeline for simplified tone-mapping algorithm.

3.3 Colorization

We are interested in colorizing images and video to add back additional contextual visual color clues lost with mono-spectral imagers, both for visible and non-visible spectra. Because research [Varga 99] shows that grayscale images colorized to resemble the color of day time images enhances the ability to perform perceptual tasks, we target natural colorization results.

Accurate colorization relies on two necessary inputs. First, a relatively noise-free version of the grayscale video is needed to prevent the colorization algorithm from confusing texture with sensor noise. Second, a training image or video is needed from which to learn a colorization model. Based on the quality and type of both the source video and training set, we have developed a variety of applicable colorization techniques.

In the first scenario, we have a training image taken from the exact camera position under daytime illumination. Our grayscale video is then taken during the night under significantly different lighting conditions. Using Poisson interpolation techniques it is possible to alter the luminances of the dark video while maintaining its details. Once the luminances resemble those of the daytime image, the daytime chrominance can be inserted, resulting in a color image that has daytime luminance and chrominance qualities.

In the second scenario, we do not have a training image that was taken with the same camera intrinsics or extrinsics. However, the training image must be similar in that it contains the general scene elements as seen in the dark grayscale image. In this case, it is advisable to use a noise reduction and tone-mapping algorithm pair, such as those presented in Section 3.1 and Section 3.2, in order to make the dark video resemble the daytime video. By doing comparisons of local image statistics of both the originally dark video and the training image, chrominance can be copied from the training set to the grayscale video. A technique such as this can run into trouble when an exact match in image statistics does not exist between the two data sets. Trouble can also arise if residual noise still exists in the dark video, making it difficult to discern texture.

In order to handle noisier video that is more unlike the training set, a third technique, based on belief propagation, is introduced. In this approach, local statistics beyond luminance and variance are used to match areas of similar visual characteristics. Furthermore, for increased robustness and coherency, local spatial consensus is reached on which area to copy chrominance from in the training data. Iterative belief propagation is used to minimize the number of target areas and to limit changes in chrominance to image regions containing luminance edges.

3.3.1 Colorization of Poisson Interpolated Images

We begin by discussing how color can be transferred from a color source image to a grayscale image taken from the same camera. Such an image pair could easily result from a daytime, visible-spectrum, multi-channel color image and a mono-channel night-vision image, both taken with a multi-spectral imaging rig. We could copy the chrominance information from the color image to the grayscale image directly, but because the nighttime luminances do not match with

the daytime luminances, the result is not desirable. Instead, if the luminances of the nighttime image resembled the daytime luminances, direct copying of the chrominance would result in a higher quality output. We match the luminance qualities of the daytime by reintegrating the gradient field of the nighttime image with the boundary conditions of the daytime image.

The gradient field of an image indicates the first-order image derivative at each point (i.e. how much a pixel's value differs from its neighbors' values). These gradients are what the human visual system interprets as edges and are crucial to image interpretation. The generating image is only recoverable from a gradient field if the boundary conditions are known. Without boundary conditions, only relative image values can be known, not absolute values.

Recovery of an image is achieved through a technique known as Poisson interpolation that has been used for fluid dynamics, thermal dynamics, and material deformation. The Poisson equations shown in equations 6 and 7 can reconstruct an image given a set of boundary conditions to guide reconstruction.

$$\min \iint \left\| \nabla \boldsymbol{I} \right\|^2 \, \partial \boldsymbol{x} \, \partial \boldsymbol{y} \qquad (6)$$

$$\nabla^2 \boldsymbol{I} = 0 \qquad (7)$$

The notion of using Poisson techniques in computer graphics has recently been of great interest. Their use in reintegrating gradient fields of HDR images to have decreased dynamic range was explored by Perez et al [03]. Soon after, Fattal et al [04] introduced the notion of using gradient space techniques to combine scene elements from multiple images by combining their gradient fields. The most pertinent publication of the topic is from Raskar et al [04], who discussed mixing gradients from both nighttime and daytime images into a new image. This new image could then be reintegrated to have the detail of day but with the new scene elements present at night. Our approach allows for temporal coherence in video processing, by enforcing stricter boundary conditions.

These recent works all create synthetic gradient fields that cannot be integrated because they lack the quality of being flux-conservative (i.e. no possible image corresponds to them). Yet, it is still possible to find the image whose gradient field is most similar to the desired gradient field in terms of least squares error. The Poisson equations are restated in equations 8 and 9 to find this minimum error solution. These Poisson methods can support Dirichlet boundary conditions and typically are solved using iterative methods.

$$\min \iint \left\| \nabla \boldsymbol{I} - \vec{\mathbf{v}}(\boldsymbol{x}, \boldsymbol{y}) \right\|^2 \, \partial \boldsymbol{x} \, \partial \boldsymbol{y} \qquad (8)$$

$$\nabla^2 \boldsymbol{I} = \boldsymbol{div} \, \vec{\mathbf{v}} \qquad (9)$$

These non-flux-conservative fields can be created in many possible ways to solve various problems. One way is to arbitrarily manipulate gradient values, such as to reduce dynamic range. Another way is to combine gradient fields from multiple images into a single gradient field. We make our nighttime image resemble the daytime images by reinterpolating them with

the boundary conditions of the daytime images. Because the boundary conditions do not match the rest of the gradient field, we are also creating a non-flux conservative scenario without ever changing the gradients themselves.

After interpolating with the new boundary conditions, the dark image becomes brighter, as shown in Figure 15. The chrominance can be directly copied from the daytime image using any color space that separates luminance and chrominance, as will be described in more detail in section 3.3.3.1.

If new scene elements are present or missing from the brightened nighttime image as compared to the daytime image, they will reflect the original scene's coloration at those pixels. If the scene element is small, generally inaccuracies are difficult to notice, but if the elements are large this becomes more noticeable. However, because the human visual system interprets luminance more than chrominance, it is still possible to perceive the image's content.

Further investigation has been done by interpolating mixed gradient fields for improved surveillance. For example, suppose you have a stationary surveillance camera which can perform rudimentary motion detection even in dark environments capturing data all day and night. Using the motion detection, the area in and around each moving object can be considered a mask. The gradient fields of a static daytime scene and the nighttime scene can be mixed based on this mask. Namely, the daytime background gradient can be used everywhere there is no motion and the nighttime portion can be used anywhere there is motion. Again using the daytime boundary conditions, the image can be re-interpolated using the same Poisson system. This process is shown in Figure 16.

Figure 15: Poisson interpolation with boundary conditions from a daytime image taken with the same camera. Upper Left: daytime RGB image, Upper Right: long exposure nighttime RGB image, Bottom Left: long exposure gradient field interpolated with daytime boundary conditions with nighttime chrominance, Bottom Right: same as Bottom left with daytime chrominance

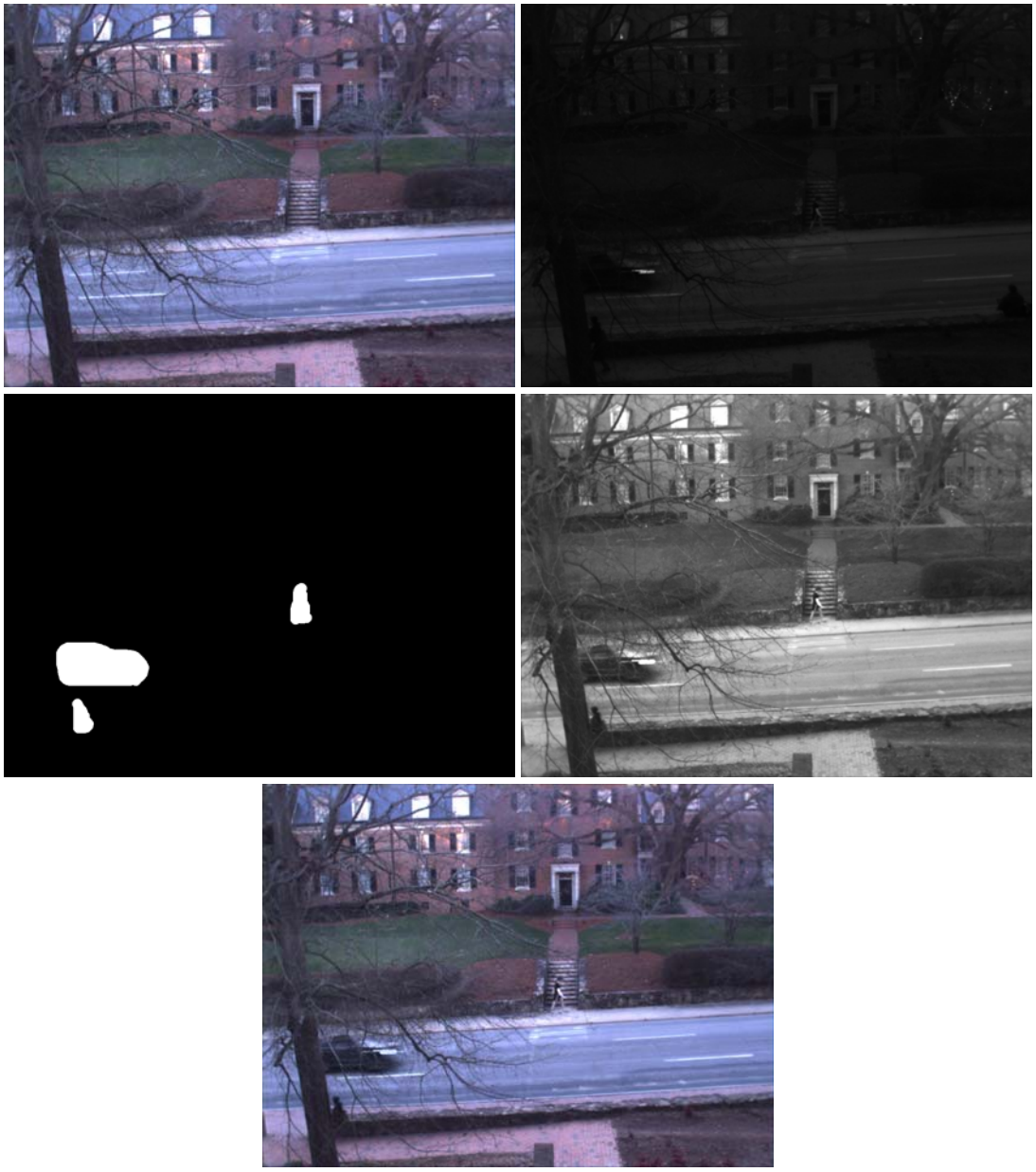


Figure 16: Illustration of mixing gradient fields from daytime and nighttime images. Upper Left: a daytime RGB image; Upper Right: a grayscale nighttime image; Middle Left: a mask displaying areas of difference between the two images; Middle Right: Poisson interpolation of the mixed gradient fields as defined by the mask; Bottom Center: Poisson interpolated image with chrominance copied from daytime RGB image.

3.3.2 Colorization Based on Mean and Variance

Welsh et al [02] propose an algorithm to colorize a grayscale image by transferring color from a source color image to a target grayscale image. Toet [05] applied this algorithm to images from traditional light-enhancing night vision systems and generated a set of promising results. To obtain these results, the source and target images contained consistent scene elements, although they were not the same image, such images including similar foliage.

This similarity is necessary because the algorithm operates by finding the closest per-pixel match between the source and target images. The quality of a match is based on the luminance statistics of the pixel's neighborhood, specifically the mean and variation. Once a match is found, the corresponding chrominance is copied from the source to the target pixel, while the target's luminance is preserved.

Although this algorithm can give some good results, it has three drawbacks that limit its usefulness:

- The algorithm randomly selects a fixed number of color samples from source image and only these samples are used in the matching process. Poor matching can result if an insufficient number of samples fall in source regions common in the target image.
- Because only the mean and variance of a pixel's neighborhood are used for matching, it may not be possible to handle areas of rich textures. Thus, more sophisticated feature modeling is needed.
- Due to the per-pixel matching scheme, only local evidence is considered and not higher-level trends. This means there can be no spatial coherence, making the algorithm sensitive to noise and outliers. This is specifically problematic in high-noise night vision images.

Most recently, Tai et al [05] proposed a regional color transfer algorithm to exploit the spatial coherence constraint. The focus of their algorithm is to segment the image into homogeneous color regions and then to use Expectation-Maximization (EM) to produce an optimal number of color regions. However, their region matching between source and target image is still based only on the mean intensity of the regions and cannot distinguish regions where the only variation is in texture.

3.3.3 Belief Propagation Colorization

As a solution to the shortcomings of prior techniques, a new framework is proposed to transfer color from color source images to grayscale target images. This framework uses a color model which is chosen based on color space behavior criteria. The colorization algorithm is then divided into three steps: source chrominance segmentation, feature model estimation, and chrominance prediction. The framework is sufficiently flexible to adapt to various applications and fully utilize the properties of the input data.

3.3.3.1 Color Space Representation

Typically, three channel color space representations are used in computer graphics because they mimic the color receptors in the human visual system (i.e. red, green, and blue). As in anatomy,

these channels are not truly orthogonal and correlations exist between different channels. Considering RGB, each channel encodes both a portion of the intensity and the chromatic information. For our purposes, a color model that represents intensity and chrominance separately would be preferred, so we can preserve luminance and add or modify chrominance.

We consider two candidate color spaces that separate chrominance and luminance: YUV and *lαβ*. YUV, commonly used in video, contains an intensity channel, Y, and two chrominance channels, U and V. *Lαβ*, proposed by Ruderman et al [98], minimizes correlations for most natural scenes. In this space, *l* represents intensity and *α* and *β* encode the chromatic information. YUV color space is well understood, while *lαβ* is comparatively new, but both have been used for colorization by previous researchers [Welsh 02] [Levin 04]. Reinhard et al [01] suggested that *lαβ* is more suitable to perform colorization because its channels are more independent; changes made in one channel have minor effects in other channels. In experiments, we found that YUV and *lαβ* usually perform similarly, and only occasionally is one superior. Thus, we chose YUV for our implementation because of its simpler color space transforms with RGB.

3.3.3.2 Belief Propagation

Our color reconstruction is formulated as a Markov Random Field(MRF) and belief propagation is applied to robustly estimate the chromatic value for each pixel. This section contains an introduction to MRFs and to the belief propagation algorithm.

An MRF is an undirected graph, as shown in Figure 17. Two kinds of nodes are in this graph: hidden variable nodes $\{x_p\}$ and observed variable nodes $\{y_p\}$. We assume there are two kinds of statistical relationships among these nodes. One is statistical dependency between the hidden variable x_p and the observed variable y_p at each position *p,* which is be represented by a joint compatible function $\phi(x_p, y_p)$. The other is the statistical dependency between every two neighboring hidden variables x_p and x_q, represented by the function $\varphi(x_p, x_q)$.

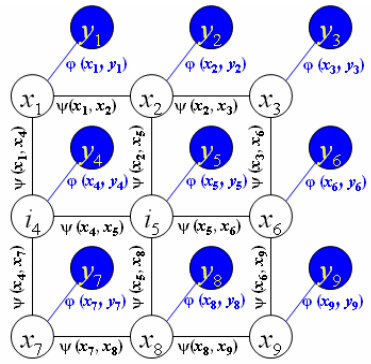


Figure 17: Illustration of the graph representation of a Markov Random Field. $\{x_p\}$ are the hidden variable nodes and $\{y_p\}$ are observed variable nodes. The statistic relations among these nodes are encoded in $\phi(x_p, y_p)$ and $\varphi(x_p, x_q)$.

MRFs are typically used to model inference problems, which estimate the hidden variables $\{x_p\}$ based on observed variables $\{y_p\}$. The local evidence of x_p given y_p, is encoded by $\phi(x_p, y_p)$ and the compatibility of neighbor hidden variables is encoded by $\varphi(x_p, x_q)$.

Let $\phi(x_p, y_p)$ be the observation probability $p(y_p \mid x_p)$. The posterior $P(\{x_p\} \mid \{y_p\})$ can be expressed as:

$$P(\{x_p\} \mid \{y_p\}) \propto \prod_p \phi(x_p, y_p) \prod_{p,q} \varphi(x_p, x_q) \qquad (10)$$

Estimating the most probable $\{x_p\}$ given $\{y_p\}$ involves finding the Maximum a Priori(MAP) solution of MRF. This means we want to find the solution of $\{x_p\}$ which maximizes equation 10. For computational convenience, this is equivalent to minimizing the negative log of equation 10, where the max-product becomes a min-sum. The negative log of equation 10 is called the energy function of the MRF and is written as:

$$E = \sum_p D(x_p) + \sum_{p,q} V(x_p, x_q) \qquad (11)$$

where $D(x_p) = -\log(\phi(x_p, y_p))$ is called the data cost function and $V(x_p, x_q) = -\log(\varphi(x_p, x_q))$ is called the compatible cost function

Usually, x_p is a discrete random variable. In this situation, finding the exact MAP solution is intractable due to the large solution space. Belief propagation [Yedidia 03] has recently been proposed to give an approximate solution and has been successfully applied to many vision problems including image super-resolution [Freeman 00] and stereo matching[Sun 03].

Belief propagation is an iterative inference algorithm that passes messages around the MRF network. Each message is a vector of dimension given by the number of possible values of x_p. Let $m_{p,q}^t$ be the message sent from a hidden node *p* to the hidden node *q* at time *t*. If belief prorogation is performed on the energy function (Equation 11), all entries in $m_{p,q}^0$ are initialized to zero, and the new messages at each iteration are updated in the following way:

$$m_{p,q}^t(x_q) = \min_{x_p} (V(x_p, x_q) + D(x_p) + \sum_{s \in N(p) \setminus q} m_{s,p}^{t-1}(x_p)) \qquad (12)$$

where $N(p) \setminus q$ represents the neighbors of *p* other than *q*.

After T iterations, a belief vector is computed for each node:

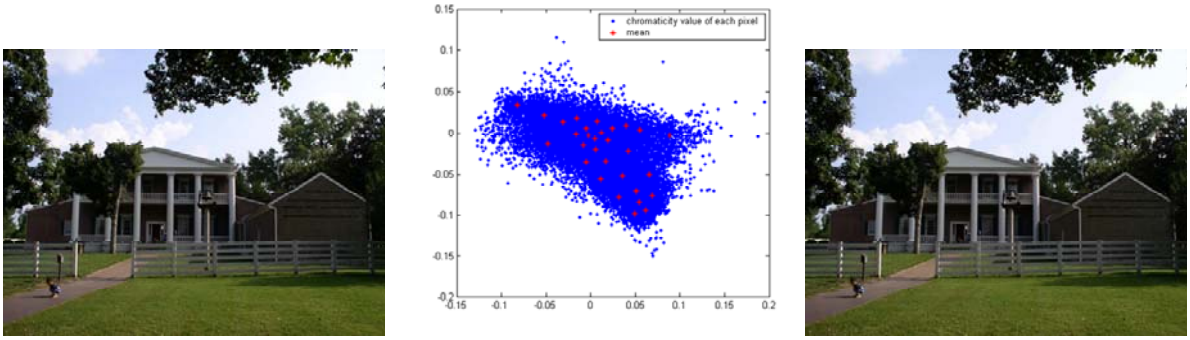
$$b_q(x_q) = D(x_q) + \sum_{p \in N(q)} m_{p,q}^T(x_q) \qquad (13)$$

For each node, the value x_q^* that minimizes $b_q(x_q)$ is chosen as the estimation result. The time complexity of standard belief propagation algorithm is understood to be $O(mk^2T)$, where m is the number of pixels in the image, and k is the number of possible values for hidden variable x_p. Recently, Felzenszwalb et al [04] proposed an efficient implementation with time complexity $O(mkT)$. In our system, we adopted this algorithm for efficiency.

3.3.3.3 Modeling Image Chrominance

One could observe that the full range of colors of a normal image can be replaced by only a small number of colors and still achieve satisfactory perceived visual quality. This is because human vision is more sensitive to intensity than chrominance, as we illustrate with the following example. The color space of a typical photo of a natural scene is initially transformed from RGB to YUV space. We plot each UV chrominance value as a point in 2D color space. A K-means clustering then finds the 30 clusters in the color space with the highest density. The means of these clusters represent the most frequent chrominances in the image. We then we replace each UV value in the original image with closest UV cluster mean, then transform the image back into RGB.

Figure **18** illustrates this process and shows that only 30 distinct chrominances represent a sufficient color distribution to recreate an image.



(a) (b) (c)

Figure 18: The color of a natural images can be represented by only a few chrominances. (a) A picture taken from natural scene. (b) The UV color channels of the image and the 30 representative chrominances values (red) obtained from K-mean. (c) The reconstructed images only using 30 representative chromatic values.

A second observation is that the chrominance values of an image have spatial coherence. This means that the chrominance of a pixel tends to be similar to its neighbors. In this manner, the chrominance of an image is a smooth field with abrupt changes that typically occur at the same

locations that intensity makes an abrupt change. This observation forms the basis for our smooth prior of the color model.

Using the above observations, we can create a statistical image color model. The colors contained in an image are represented by the set of n chromatic values $C = \{c_1, c_2, ..., c_n\}$. Each pixel is assigned to have an associated C value and pixels that share the same C value belong to the same class R. It follows that there exist n classes $R = \{R_1, R_2, ..., R_n\}$, one for each C. All pixels in a class R_i tend to be spatially clustered. Conceptually, we are saying images are composed of blocks of similar chrominance.

This model can be formulated as a Markov Random Field(MRF), which in turn can be used to robustly estimate the chromatic value of each pixel. The energy function associated with our MRF is:

$$E = \sum_{(p,q)\in N} V(c(p), c(q)) + \alpha \sum_{p \in P} D_p(c(p)) \qquad (14)$$

P is the set of pixels in an image. N is the set of the neighborhoods around every pixel. Normally, the 4-connected neighborhood set is used. $c(p)$ is the chrominance value assigned to pixel p, where $c(p) \in \{c_0, c_1, ..., c_{n-1}\}$. $D_p(c(p))$ is the cost(local evidence) of assigning $c(p)$ to pixel p, which is referred to as the data cost. $V(c(p), c(q))$ is the cost of assigning $c(p)$ and $c(q)$ to two neighbor pixels, which is referred to as the compatible cost. α is the coefficient which determines the relative importance between data cost and compatible cost. Finding the assignment of $c(p)$ with minimum energy of E corresponds to the MAP estimation of the MRF. Belief propagation [Yedidia 03] can be used to solve this problem.

The smooth prior of the color model can be expressed as the compatible cost:

$$V(c(p), c(q)) = \beta * \delta(c(p), c(q)) \qquad (15)$$

where δ is the indication function:

$$\delta(x_1, x_2) = \begin{cases} 1 & \text{if } x_1 \neq x_2 \\ 0 & \text{otherwise} \end{cases} \qquad (16)$$

We assume the chromatic values should be piecewise constant. If the chromatic values of two neighboring pixels are different, a penalty will be incurred. The severity of this penalty is determined by the variable β. As mentioned, the spatial change of chromatic values often coincides with the spatial change of the intensity. Therefore, if the intensities of two neighboring pixels are similar, the chrominance of those pixels should also be the same. We modulate our penalty strength to reflect this:

$$\beta = \exp(-(I(p) - I(q))^2 / 2\sigma_I^2) \qquad (17)$$

$I(p)$ and $I(q)$ are the intensities of p and q. σ_I determines how local intensity smoothness affects the chrominance smooth prior. Smaller σ_I values force the chrominance smooth prior to be weaker at intensity edges. Figure 19 shows the relationship between β and $I(p) - I(q)$ with different sigma σ_I values.

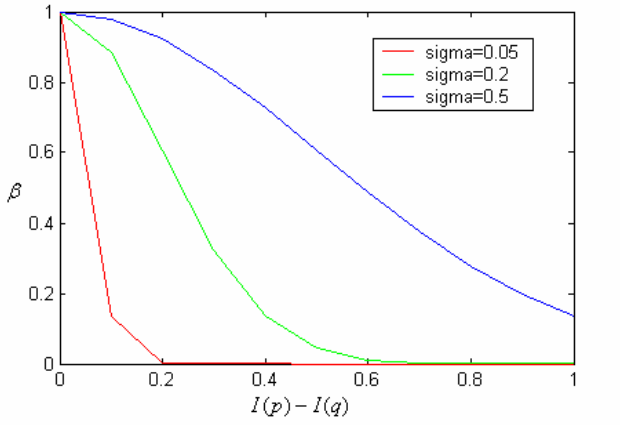


Figure 19: Plot of the function β of $I(p) - I(q)$. Smaller σ_I values indicate the smooth prior of chrominance will be weaker at the intensity edges.

3.3.3.4 Algorithm Overview

In example based colorization, color is transferred from source color images to target monochromatic images. In this discussion, we only consider a single source image. However, it is straightforward to extend our algorithm to support multiple source images. We assume that the properly colorized target images will have the same color characteristics as the source image.

We formulate the problem with our proposed color model. Let I_s and I_t be the source and target images respectively. With a set of n chrominances $C = \{c_0, c_1, \ldots, c_{n-1}\}$, I_s and I_t can both be segmented into n classes, which will be discovered in the course of the algorithm. We will colorize by verifying that pixels in the color class R_i^s in I_s match those pixels R_i^t in I_t, with some metric of visual similarity.

The colorization process can be divided to three steps: chrominance segmentation, feature model estimation, and chrominance prediction. Chrominance segmentation and feature model estimation can be viewed as uncovering the visual qualities of regions of similar color in the source color image. Chrominance prediction uses these visual qualities to transfer color to the target images.

3.3.3.5 Chrominance Segmentation

At this step, we want to find the n representative chromatic values $C = \{c_1, c_2, \ldots, c_n\}$ in the source image. Based on these chromatic values, all image pixels can be partitioned into n classes, $R = \{R_1, R_2, \ldots, R_n\}$.

The first thing to determine is the number of chromatic values, n. n should be chosen to give an accurate representation of the chrominance space of the source image. If n is too small, it can not sufficiently sample the chrominance space. However, if n is too large, the number of pixels belonging to one chrominance region is not sufficient to characterize its visual properties. This over-segmentation will cause problems in feature model estimation. Finding the optimal n would involve a complete model of human color perceptual tolerance and that is a difficult problem. Fortunately, algorithm performance is not sensitive to this parameter when n is within a proper range. For our experiments, n is always set to 30.

To find n representative chrominance values, K-means clustering is used to group the chromaticity of the source image into n number of groups. K-means is an unsupervised clustering algorithm which minimizes the sum of squares of distances between data and its associated cluster centroid. The centroid of each group will be used as the representative chrominance value of this group.

After K-means, each pixel of the source image must be labeled with one of the representative chrominance values. The simplest way is to use the value closest to the original chrominance. However, because of sensor noise and the ambiguity that can result when choosing from between multiple similar representative values, the chrominance regions will not satisfy the smooth prior. We want to avoid having isolated pixels whose representative chrominance values are different from all of their neighbors.

Equation IV.5 is used to enforce the smooth prior of the chromatic regions. For this portion of the discussion, the intensity values used in equation IV.8 are the intensities of the source image. The data cost term describes the distance between each representative chrominance and its original chrominance.

$$D_p(c(p)) = \min(\left\| c(p) - c^o(p) \right\|, T_1) \qquad (18)$$

where $c(p)$ is the representative chrominance assigned to pixel p and $c^o(p)$ is the original chrominance. This function is robust to the effects of outliers and noise. Furthermore, its maximum output will be bounded by the threshold T_1.

3.3.4 Feature Model Estimation

After segmentation, the source image is partitioned into chrominance regions. Based on these regions, the intensity channel will be analyzed to build a feature model which characterizes each region. The identifying feature vector of each pixel is calculated based upon the intensity at its neighbors. A statistical model is estimated for each chrominance region based on its component feature vectors' statistical properties.

Choosing which statistical features should compose the feature vector is determined by how similar the target image is to the source image. If the relative lighting conditions of the source and target images are similar, the mean and standard deviation of the pixel's local patch can be used as a feature vector. In previous research, Toet used this feature vector to do colorization. If the source and target image come from the same surveillance camera, the positions of the source and target pixels are closely correlated. Thus, a pixel's spatial position can be included in the feature vector. If the source and target image have similar texture (e.g. grass) the filter bank [Malik and Perona 90] will convolve the image to get the feature vector.

The filter bank is a set of orientation and spatial-frequency selective linear filters. The filter responses are used to represent the texture property of an image. The filters we use are proposed by Malik et al [01], including elongated Gaussian derivative filters with 2 phases, 6 orientations, 3 scales and center-surround Gaussian derivative filters with 4 scale. Figure 20, which comes from the original paper [Malik et al 01], illustrates this filter bank.

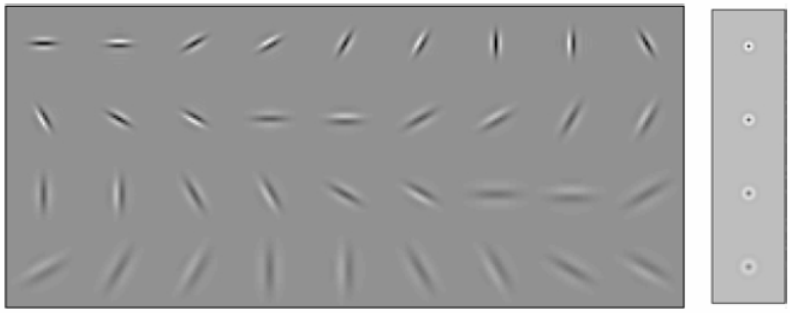


Figure 20: The filter bank used in texture analysis [Malik 01]. The left ones are elongated Gaussian derivative filters with 2 phases, 6 orientations, 3 scales. The right ones are center-surround Gaussian derivative filters with 4 scales.

We support all three of these feature vectors and choose from them based on the properties of the input images.

Suppose τ_p is the feature vector of pixel p. Within each chrominance region R_i, we have the feature vector set $\{\tau_p\}_i = \{\tau_p | p \in R_i\}$. A statistical model ϕ_i is estimated from $\{\tau_p\}_i$ to describe the distribution of $\{\tau_p\}_i$. The model can be parametric, such as Gaussian mixture, or nonparametric, such as kernel density estimation. Parametric models typically need prior knowledge of the sample distribution, but nonparametric models can give good performance without prior knowledge of the sample distribution. For parametric models, the only unknown quantities are a finite set of constant parameters that can be estimated from the data. Nonparametric models can estimate the probability function directly from the density of the sample distribution. It follows that the computational cost of using a nonparametric model is higher than for a parametric model. Nevertheless, we choose to use the nonparametric model.

Let $p(\tau \mid \phi_i)$ be the probability that feature τ belongs to class *i* given the model ϕ_i:

$$p(\tau \mid \phi_i) \propto \exp(-\sum_{t=1}^{k} \left\| \tau - \tau_p^t \right\|) \qquad (19)$$

where $\{\tau_p^1,...\tau_p^k\} \subset \{\tau_p\}_i$ and $\{\tau_p^1,...\tau_p^k\}$ are K -nearest feature vectors to τ. The value of K determines how many local samples are used to estimate the distribution density.

3.3.5 Chrominance Prediction

The colorization process can now use the learned model of the source images to locate and transfer color to the target grayscale image. The chrominance of the target image is predicted from the estimated feature model. For robustness estimation, this is done using the constraints of the color model of equation IV.5. Here, the compatible cost is calculated based on the intensity of the target grayscale image and the feature vectors are derived from the target image. The data cost becomes:

$$D_p(c(p)) = \min(-\log(p(\tau_p \mid \phi_{c(p)})), T_2) \qquad (20)$$

where $p(\tau_p \mid \phi_{c(p)})$ is the possibility that the feature vector τ_p belongs to the class $c(p)$. As part of the formulation, the maximum output is guaranteed to be T_2.

3.4 Luminance Transfer

In the colorization algorithms discussed in Section 3.3, the ability to colorize SWIR video was hindered because the relative luminance intensities of the visible spectrum do not always correspond to the relative luminance intensities of the SWIR domain. Thus, when chrominance is transferred, it can look unrealistic due to having correct colors blended with incorrect luminances.

Because our goal continues to be the processing of night-vision imagery to resemble daytime photographs, we address this issue. Basically, a luminance transfer algorithm is needed which operates in a similar manner to the chrominance transfer algorithm in Section 3.3.3. However, chrominance and luminance are two very different variables, and the human visual system can much more easily notice errors in luminance than in chrominance. When dealing with chrominance, the human visual system can tolerate incorrect colorizations due to mismatched regions in the source example images and the target images. These mismatches, however, are noticeable and intolerable for luminance transfer. This makes simple, per-pixel luminance transfer infeasible.

One way to solve this problem is to apply a global luminance adjustment to the target image. Histogram matching is normally used in image processing to make a source image and a target image have same first order luminance statistics. Recently, a simple and effective method proposed by Reinhard et al [01] tried to make the target image have similar appearance by globally adjusting the target image to have the same mean and variance as the source image.

While global adjustment is quite effective in some applications, it does not satisfy our requirements. This type of algorithm can make the luminance of one region brighter or darker, but cannot change the relative luminance of different regions. In night vision IR images, parts of the image, such as trees, grass and living creatures, are much brighter than their surrounding regions. If we want to create a daytime appearance, the relative luminances need to be changed. Some dark regions need to be brightened, while some bright regions need to be darkened. Therefore, this cannot be archived through global adjustments or linear filtering.

Our approach is to apply a region-based luminance adjustment to the target image. The target is first partitioned into regions. All pixels in each region will receive the same mean and variance adjustment. The magnitude of the adjustment is determined by examining matching regions in the source images. This way, we preserve relative luminances within regions, but can alter relationships between regions. This enables us to modify nighttime IR luminance images to look like daytime images while avoiding spatial coherence issues caused by per-pixel alteration.

3.4.1 Algorithm Overview

The goal of our algorithm is to modify the luminance of nighttime IR images based on an example. The example is a pair of registered source images, a nighttime IR image and a corresponding daytime RGB image. The pair of image can be obtained by capturing images at

the same location during the day with an RGB camera and at night using an IR camera. Then, for a target nighttime IR image from a similar scene, the luminance is adjusted to make the appearance similar to the luminance of the training pair's RGB image.

The luminance of the target IR image is modified locally based on the properties of each region containing similar texture. We want to make all pixels for one kind of texture in the target image equally darker or brighter. For each texture region in the target IR image, the algorithm searches for the most similar region in the source IR image. Then, the mean and standard deviation of the luminance of the region is adjusted to match the characteristics of the source region.

More concretely, if *l(p)* is the luminance of a pixel in the target image, we modify it as such:

$$l(p)' = \frac{\sigma_s^{R_t}}{\sigma_t^{R_t}}(l(p) - \mu_t^{R_t}) + \mu_s^{R_t} \qquad (21)$$

where R_t is the region to which pixel p belongs and $\mu_t^{R_t}$ and $\sigma_t^{R_t}$ are the mean and standard deviation of luminance, respectively, of the target region R_t. $\mu_s^{R_t}$ and $\sigma_s^{R_t}$ are the mean and standard deviation of luminance calculated from the source images pairs.

The main function of the algorithm is to find corresponding regions in the source image pair for each target region, and calculate the luminance parameters $\mu_s^{R_t}$ and $\sigma_s^{R_t}$.

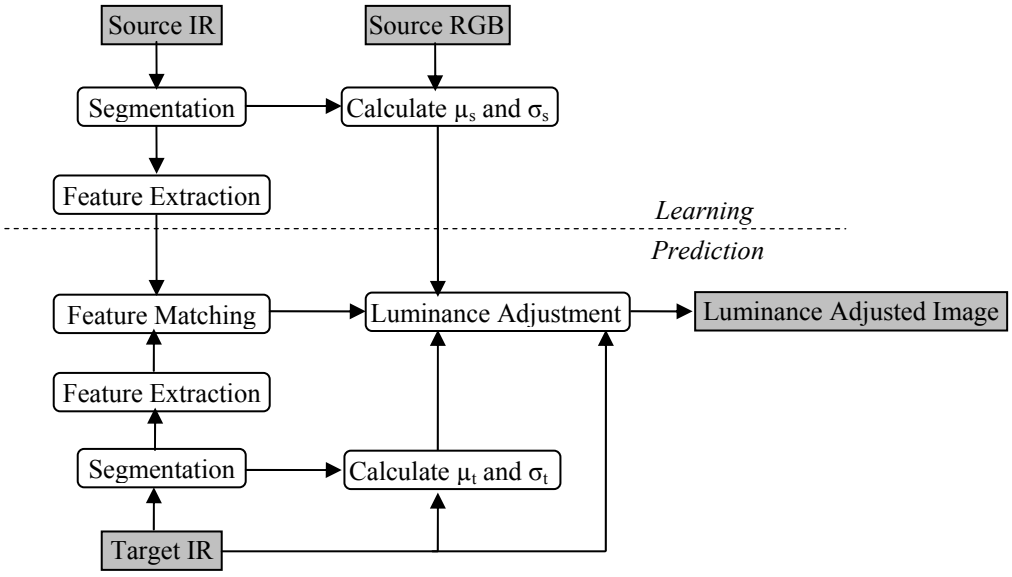


Figure 21: A flowchart of the entire process for luminance adjustment.

The flow chart of the algorithm is shown in Figure 21. The algorithm can be divided into two stages: learning and prediction. The learning stage analyzes the source image pair to find image statistics for each region, including the mean and deviation of luminance. This portion of the algorithm can be treated as an off-line preprocess. In the prediction stage, the target IR image is then analyzed and the luminance modified based on Equation 21.

Specifically, the learning stage includes three steps. First, the source IR image is segmented into homogenous texture regions. Second, a feature vector is extracted from every region. This feature vector represents the texture properties of each region and differentiates it from other regions with different textures. Finally, as mentioned before, the mean and deviation of luminance of each region in the RGB source are calculated.

The prediction stage includes five steps. The first three steps are similar to the learning stage. The target IR is segmented and feature vectors are extracted, this time using statistics from the target IR image. Then, for each region in the target IR image, we find the similar regions in the source IR images based on similarity between feature vectors. Finally, the luminance qualities of the matching regions are applied to the target region.

There are three major technology components necessary to implement the learning and prediction stages: segmentation, feature extraction and matching, and luminance adjustment. The details of these components are discussed in the following subsections.

3.4.2 Segmentation

Because we need to obtain the image statistics and then uniformly modify the luminance characteristics of regions of like texture in IR images, we need a way to accurately segment both the source and target IR images. By finding accurate edges, we can ensure that changes in relative luminance will be applied to entire objects with modifications bounded by object edges.

Certainly, the quality of the final luminance adjustment is heavily influenced by the quality of segmentation results of both the source and target IR images. To obtain good luminance adjustment results, segmented regions should be of homogenous texture and luminance. Two areas with different desired luminance levels should not be included in the same region. For example, if the relative luminance of the sky and the building needs to be reversed, the segmentation algorithm must be able to separate these two regions. Note that we can tolerate some degree of over-segmentation. Segmenting the same kind of texture into two regions will not influence the final results, as long as the luminances of these two regions are modified similarly. Successful feature matching will guarantee this to occur.

Image segmentation has been the subject of considerable research over the last three decades. Many segmentation algorithms have been proposed which use different cues, such as luminances, edges, and textures. A good survey and evaluation of these algorithms can be found in [Zhang 97]. For our purposes, we have chosen the mean-shift segmentation method proposed by Comaniciu et al [02] due to its robustness and ease of control (with only three parameters). This algorithm uses mean-shift analysis to cluster local features, including luminance, gradients and the filter response. We let the algorithm over-segment the image, but to prevent too many regions, we set the smallest region size to include at least 400 pixels. The same set of segmentation parameters are used for both source and target IR images to capture the same scale of segments.

3.4.3 Feature Extraction and Matching

For each region, a feature vector is extracted to describe its texture properties. A good feature space is desirable if it can distinguish between different kinds of textures in the IR images. The design of a feature space involves two choices[Chen et al 98]. The first choice is what kind of filter should be used to analyze the feature of the each pixel's neighborhood. Widely used filters include Gabor filter banks, wavelet transforms, quadrature mirror filters, and discrete cosine transforms [Randen et al 99]. The second choice is what kind of statistical model should be used to describe local texture features. Common choices are first and second order statistics. First order statistics can be computed from the histogram of pixel coefficients in each region. They depend only on individual pixel values and not on the interaction or co-occurrence of neighboring pixel values. Second order statistics describe the properties between pairs of pixel values, using analysis constructs such as co-occurrence matrices.

We use an approach similar to that proposed by [Van de Wouwer et al 99]. In that method, the histogram of wavelet coefficient amplitudes is used to describe the texture of each region. Wavelet transforms are used because of their wide availability and computational efficiency. Another benefit is that they represent texture properties in a multi-scale manner, which allows the same texture at different scales to generate corresponding histograms. We believe these histograms are sufficient to describe texture property in each region, because wavelet transforms capture the characteristics of local inter-pixel spatial relationships.

In our implementation, a Non-Decimated Wavelet Transform (NDWT) [Nason et al 95] is calculated for the IR images. Like the traditional wavelet transform, DWT, it applies the transform at each point of the image, saves the detail coefficients, and then uses the low-frequency coefficients for the next scale level. The only modification is that the subsampling step is omitted so the number of coefficients does not diminish from level to level. By using the coefficients from all levels, the texture property is effectively described at different scales.

For each wavelet decomposition level, three coefficient arrays are generated (Horizontal, Vertical and Diagonal). We calculate histograms for each region based on each of the coefficient arrays. The feature vector for a region is then obtained by concatenating the histograms of all decomposition levels. The final low frequency image of the wavelet transform is discarded, because there is no texture information remaining in this image.

When matching, each region in the target IR image will search for the most similar region in the source IR image. The Euclidean distance between feature vectors measures the similarity of the two regions. A small distance between two vectors corresponds to a high similarity.

3.4.4 Luminance Adjustment

Using segmentation and matching, the luminance parameters in Equation 21 can be calculated. $\mu_t^{R_t}$ and $\sigma_t^{R_t}$ can be obtained directly from region R_t in the target image. In this section, we consider the calculation of $\mu_s^{R_t}$ and $\sigma_s^{R_t}$.

Due to over-segmentation, one region R_t in the target image may correspond to several regions with same texture in the source image. The means and deviations of the luminances in the source RGB image in these regions may be different only because of the uneven illumination distribution. If $\mu_t^{R_t}$ and $\sigma_t^{R_t}$ are determined only by the most similar source region, undesirable block artifacts can result. This happens if two target regions contain the same texture, but differ via slowly varying luminances, and are thus classified into different regions. Thus, when correcting relative luminances, the errors will be noticeably wrong. We can alleviate this problem by calculated $\mu_t^{R_t}$ and $\sigma_t^{R_t}$ from the weighted sum of *k* similar regions:

$$\begin{cases} \mu_t^{R_t} = \sum_{i=1}^{k} (d_{i,t} * \mu_s^{R_{si}}) / \sum_{i=1}^{k} (d_{i,t}) \\ \sigma_t^{R_t} = \sum_{i=1}^{k} (d_{i,t} * \sigma_s^{R_{si}}) / \sum_{i=1}^{k} (d_{i,t}) \end{cases} \tag{22}$$

where $R_{s1}...R_{sk}$ are the *k* most similar regions in the source image and $d_{i,t}$ is the Euclidean distance between the feature vector of region R_t and the feature vector of R_{si}. $\mu_s^{R_{si}}$ and $\sigma_s^{R_{si}}$ are the mean and standard deviation of luminance, respectively, for the RGB image for region R_{si}. We further reduce block artifacts by smoothing the parameters used for transferring luminances near boundaries between regions.

3.5 Multi-Spectral Bilateral Fusion

Video captured using RGB sensors appears natural to human viewers because it captures the luminance responses of the world using the same wavelengths as the human eye. However, video capture systems in this wavelength typically fail in low-light and night-vision situations due to severe underexposure. To achieve sufficient sensor response, long exposure times must be used, which are impractical for video applications with moving objects.

Alternately, currently available sensors in the Short Wave Infrared (SWIR) and Near Infrared (NIR) spectra do not require long exposures, making them ideal for video capture. However, because they capture in a non-visual spectrum, the relative luminance they capture can vary from the visual spectrum, giving them have a disconcerting and often ghostly appearance.

In this section, we discuss how information from these two sources can be fused into a single video stream that contains both the proper luminance response and also sharp motion edges. We have already shown in prior sections that Poisson interpolation allows the mixing of gradient fields from multiple images or videos to contain the desired qualities of both. One such method is to use the boundary conditions of a well exposed image and the gradient field of an underexposed image, and then interpolate to find a new result. Another method is to specify a mask that selectively mixes the gradient fields of two videos into a new field, which is again interpolated.

Although these Poisson methods yield plausible results, there are two problems that arise in implementations. First, Poisson methods are commonly solved using iterative methods that can require many passes to complete. Although fast multi-grid solutions do exist, these methods internally require multiple iterations to converge to a solution. This time-to-convergence issue is especially prevalent in the boundary condition substitution case, where changes on the boundary need to be propagated inwards to the entire image to reach convergence. The second problem is mask specification, which can be automated using motion detection, but can be confounded by prevalent sensor noise.

To non-iteratively approximate results similar to those from Poisson interpolation and also remove the need for mask specification, we discuss the use of the joint bilateral filter as a means to mix the contributions of sensors of varying spectra. The joint bilateral can non-iteratively perform these fusion operations without the need for human interaction or mask specification. The joint bilateral separates each video source into its large scale features and its detail features. The large scale features contain the relative luminance regions, but the detail features contain the textures. The system output is a mix of large scale features of the RGB which, after noise reduction filtering, are assumed to be intact, and the detail features of the SWIR video.

In the following sections we discuss prior attempts to fuse imagery that are related to our method. We will then discuss the algorithmic details of the joint bilateral filter and describe the optical sensor configuration used to capture data. Results are presented in Section 3.6.4.

3.5.1 Related Work

The concept of fusing multiple images or videos has taken many approaches, but we will focus on works targeted at dark or night-vision enhancement.

As always, the first problem to overcome is the overwhelming contribution of noise in night-vision sensors, which obscures the texture detail content we intend to replace. Simple frame averaging for noise reduction is effective for static scenes, but creates ghosting artifacts when used with dynamic scenes. The ASTA algorithm [Bennett and McMillan 05] provides a method to reduce sensor noise without causing ghosting. However, due to its adaptive nature, it switches to spatial filtering in areas of significant motion, with the rationale being that reduced texture features are preferable to very noisy features in moving objects, where they are less likely to be noticed. So, our details instead come from IR (SWIR or NIR) video.

At the core of our fusion technique is the capability to separate the detail features from the large scale features of an image and then mix features between spectra. This is very much akin to the goal of High Dynamic Range (HDR) compression, where the dynamic range of the large scale details is decreased, whereas the relative value of the details must be preserved, necessitating an initial separation. Retinex theory [Jobson et al 97] presents a linear filtering approach to the problem by using Gaussian-derived filters, but without the use of multiple Retinex scales, sharp edges are often misclassified. The bilateral filter [Tomasi and Manduchi 98] presents a non-linear method that filters an image into regions of uniform intensity with sharp edge preserved between regions. Durand and Dorsey [02] used this separation capability to tonemap HDR images, and also presented a method to accelerate bilateral filtering through filter decomposition.

To verify that separation between detail and large scale features is done consistently between the two videos, we choose to use the joint bilateral filter proposed by Petschnigg et al [04] and by Eisemann and Durand [04] (who referred to it as “the cross bilateral filter”). Both of these papers consider the problem of combining details captured with the use of a flash with the look of an ambient image. Neither paper discussed video applications or the possibility of differing relative luminances, but they did address the issue of shadows. It should be noted that the Eisemann and Durand [04] paper is the source of our “detail” and “large scale” feature nomenclature as well as the color space used for recombining chrominance in the final algorithm stage.

3.5.2 Algorithmic Details

Processing and fusing the multiple video streams entails first decomposing each into its component large scale features and detail features and then combining the large scale features of the RGB video with the detail features of the IR to form the output. A condensed work flow diagram is depicted in Figure 22.

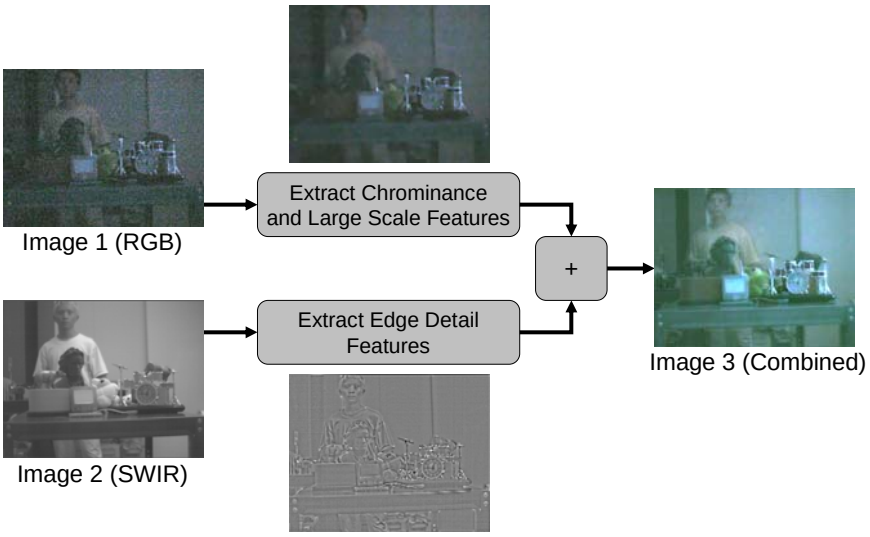


Figure 22: An overview of the multi-spectral bilateral fusion process. Each video is split into its component images, and then recombined into a noise-reduced, sharper output.

For this system, we target co-located RGB and NIR/SWIR sensors. Figure 27 shows two possible configurations of the system. The preferred method is using a beamsplitter that is reflective to the visible spectrum and transparent to all higher wavelengths. This setup reverses the reflected image, but more closely simulates the two cameras having the same optical centers, as shown in Figure 23. The other choice is to place the cameras close to each other along parallel optical paths. This setup introduces the possibility of varying occlusions between cameras, so only objects in the distance can be photographed without significant occlusion differences. In both cases, fine tuning registration is needed and is performed using a projective least-squares best-fit. Furthermore, we assume that the videos we are processing are captured at the same frame rate. The exposure time may or may not be the same between the cameras, as IR sensors are typically more sensitive than visual spectrum sensors.

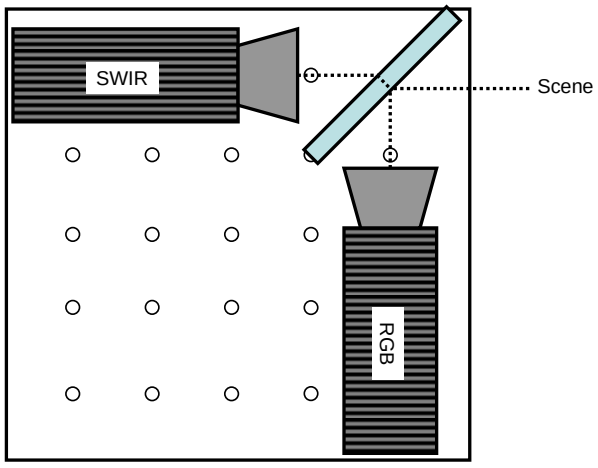


Figure 23: Optical bench setup for capture of simultaneous RGB and SWIR video streams using a cold-mirror beamsplitter. This configuration gives similar optical paths to both cameras, but a horizontally inverted image in the RGB camera. Our primary setup used a Sony DFW-V500 RGB camera and a Sensors Unlimited SU320MX SWIR camera.

Both the process of extracting large scale features and extracting details use bilateral filtering. Bilateral filtering removes small local fluctuations, resulting in areas of similar luminance separated by sharp edges. We refer to these as large scale features, which are not solely low-frequencies because of the high frequency edges. The differences between the large scale image and the original image are called the detail features, and are obtained through simple subtraction.

3.5.2.1 Joint Bilateral Filter

The spatial bilateral filter, shown in Equations 23 and 24, separates the large scale features from the detail features. For well exposed, noise-free video, such as SWIR, this technique works well, using the formulation shown below and described in Section 3.1.3.1.

$$B(s,\sigma_h,\sigma_i)=\frac{\sum_{p\in N_s} g(\|p-s\|,\sigma_h)g(D(p,s),\sigma_i)I_p}{\sum_{p\in N_s} g(\|p-s\|,\sigma_h)g(D(p,s),\sigma_i)} \qquad (23)$$

$$g(x,\sigma)=e^{\frac{-x^2}{2\sigma^2}}/(\sigma\sqrt{2\pi})$$

$$N_s=Kernel=\begin{bmatrix} p_x=[s_x-k,s_x+k] \\ p_y=[s_y-k,s_y+k] \end{bmatrix}$$

where $$D(p,s)\equiv I_p-I_s \qquad (24)$$

For noisy footage, such as that from low-light RGB or night-vision sensors, there exists no single value of σ_i that can perform sufficiently good noise reduction and large scale/detail separation. So, we need to increase the robustness of our dissimilarity function in Equation 24.

Although the relative luminances of the IR footage are not correct, the edges are properly aligned, meaning that those edges can be used in place of the noisy edges in the RGB video. Thus, all choices made by the bilateral to preserve RGB edges can be guided by the IR video. This concept is called the joint bilateral filter [Petschnigg et al 04] [Eisemann and Durand 04] and requires only the subtle change shown in Equation 25.

$$D(p,s)\equiv I_p^{IR}-I_s^{IR} \qquad (25)$$

Making this change guides the filtering of the RGB video by the IR video's edges. Thus, regardless of the intensity differences in the RGB video, which may be spurious due to considerable noise, filtering will occur based only on luminance differences in the cleaner IR source.

3.5.2.2 Video Fusion

The pipeline of our processing is shown in Figure 24. Note that we do not use ASTA filtering in this pipeline, because we can sufficiently filter the large scale features with the joint bilateral

process (and the details that ASTA would restore would be discarded). Depending on sensor characteristics, such as the presence of spatially-correlated non-Gaussian-zero-mean noise from local sensor temperature fluctuations, it may become necessary to run the joint bilateral in a 1D temporal manner as well. This removes any frame-to-frame variations visible in flat areas.

Our method can also be functionally described, as in Equation 26, where I, LS, and D are the images, large scale features, and details, respectively:

$$\begin{aligned} I_{RGB} &= LS_{RGB} + D_{RGB} \\ I_{IR} &= LS_{IR} + D_{IR} \\ I_{Out} &= LS_{RGB} + \alpha D_{IR} \end{aligned} \qquad (26)$$

The magnitude of detail features between the IR and the RGB videos is often different, due to the different sensor technologies and the relative sensitivity in low-light conditions. Thus, the scaling factor α is introduced to normalize the IR Details to be of compatible magnitude to the large scale RGB features. Experimental results show that a value of .25 to .5 often yields acceptable results between our Sony DFW-V500 and Sensors Unlimited SWIR cameras.

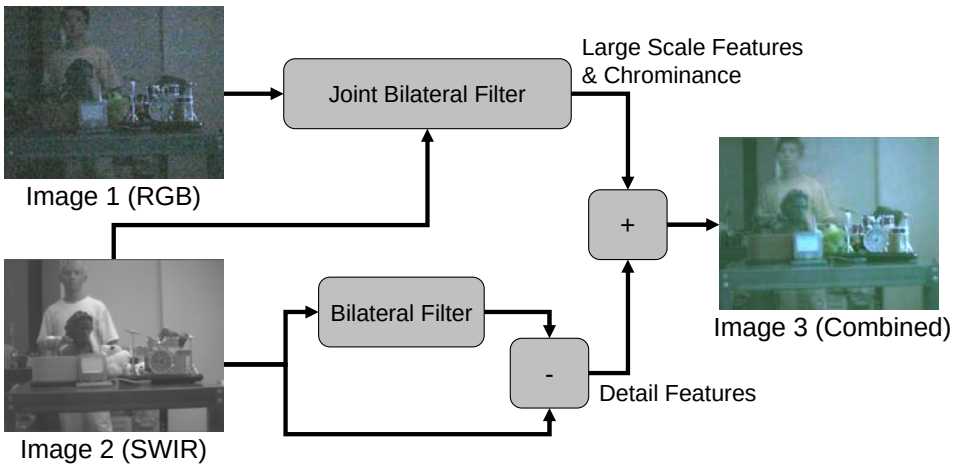


Figure 24: Detailed view of the multi-spectral bilateral fusion data flow.

Once all of the processing is complete for the luminance channel, we perform a pixel-to-pixel chrominance transfer from the original footage. Since the video containing color is assumed to also be the noisiest video, a noise reduction pre-process on the chrominance is recommended. Because the human eye is not sensitive to chrominance variations in high frequency areas, the chrominance can be processed in the same manner as the large-scale details (i.e. using the joint bilateral filter). Once the noise is removed, chrominance can be transferred using the technique described in Eisemann and Durand [04].

3.5.2.3 Log Domain Processing

Although not specified above, the entire multi-spectral bilateral fusion process is done in log space because it is necessary to process relative luminances, not absolute luminances. The human visual system is tuned to observe contrast, or how unlike two luminances are. A typical definition of contrast between two luminances A and B is (A-B)/(A+B). Thus, a luminance difference between A and B has different contrast depending on the average brightness of A and B. Operating in log-space provides that when two values are subtracted from each other, the resulting value is directly related to the contrast between those values, as shown in Figure 25.

By performing either the bilateral or joint bilateral filter in log-space, we can specify dissimilarity sigma values in terms of contrast between two pixels as opposed to their absolute values. This means that the same magnitude texture in a bright area could be categorized as a detail feature but be categorized as a large scale feature in a dark area. Furthermore, when we subtract the log-large scale features (from the bilateral filter) from the original log-image, this is the same as linear-space division. Later, when recombining features in log-space via addition, this is modulating the details by the local luminance levels. This way, a detail feature that specifies a 5% increase in the original signal will always be a 5% increase, no matter what luminance level it is placed upon. Our pipeline is shown again, in Figure 26, with all log-space/linear-space conversions indicated.

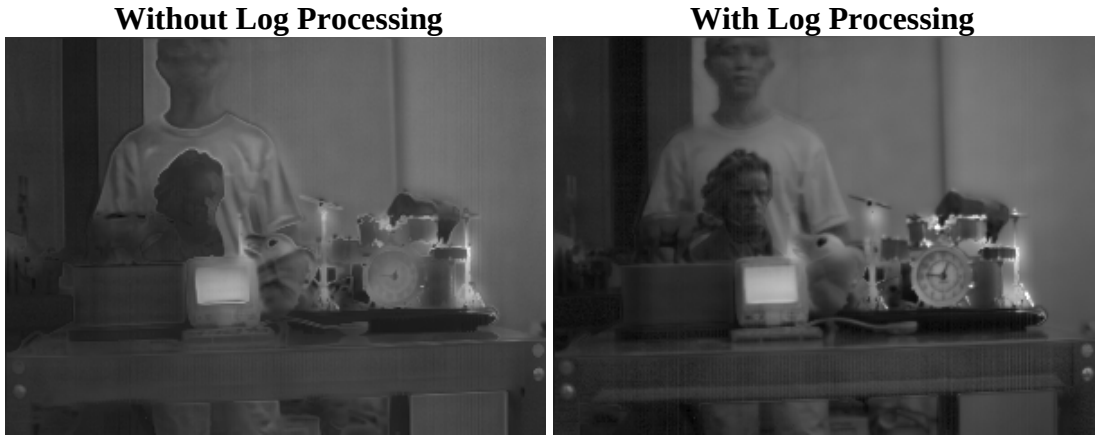


Figure 25: Depiction of the advantage of processing in the log-domain (luminance-only is shown). Without log-domain processing, the face is obscured, the details on the shirt are incorrect, the clock face is missing, and ringing is present, whereas they are properly reconstructed with log processing

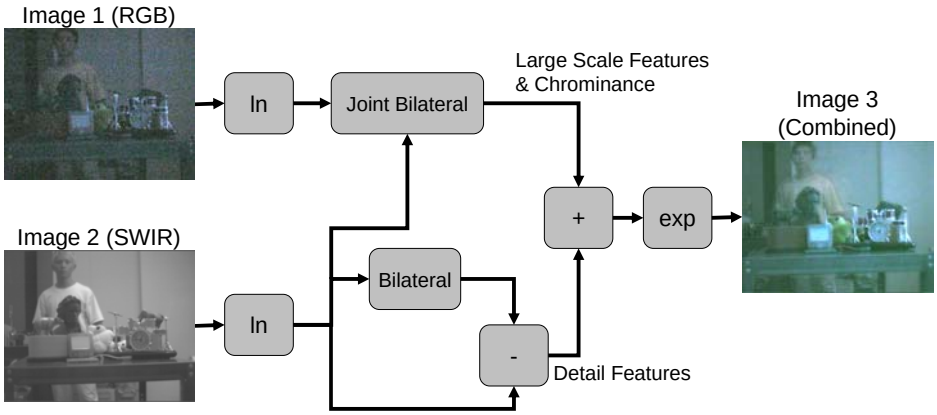


Figure 26: Detailed view of the multi-spectral bilateral fusion technique including details of the conversion to and from log-space to achieve relative luminance processing.

3.6 Results

In this section, we show and discuss the results for each of the applications of our algorithms. First, we will give an overview of the equipment used to generate our test sets.

All of the algorithms discussed in this section were tested on a variety of sensors, targeting various applications. A Sensors Unlimited SWIR imager at 320x240 was used to generate most of the grayscale videos in this section, in order to recreate the state-of-the-art in night-vision technology. In Figure 27, the SWIR imager is shown in our two capture rig configurations with the Sony DFW-V500, an uncompressed RGB 4:2:2 color imager with a resolution of 640x480, typical of surveillance equipment. Using a beamsplitter, we were able to capture data from nearly the same optical center for both cameras and then use image registration to correct the subtle differences. High-speed grayscale footage was captured from a Point Grey Research Dragonfly Express operating at 120 frames per second at 640x480. Higher resolution cameras were also tested, including the Point Grey Color Flea, at 1024x768, and ElectrIm's most recent at the same resolution, representing CCD and CMOS technologies, respectively.

Figure 27: Images of the optical bench (upper left) and field capture rigs used to capture visual spectrum and SWIR data. For short range capture, an IR beamsplitter was used. For long range capture, the sensors were placed closely together. Registration was used in both cases.

3.6.1 Video Enhancement: ASTA and Tone-Mapping

Although ASTA can be run independently of the tone mapping algorithm, its results are most commonly enhanced by performing tone-mapping on the noise reduced version to accentuate the features of the darker scene elements. Thus, visual results for both algorithms are presented here. Note that the temporal noise removal cannot be seen from still images, although it is the most prevalent type of noise in these videos and the type of noise we primarily target.

We begin by looking at nighttime SWIR video footage of a man walking through a forest in Figure 28. The video is initially too dark, and simply applying signal gain will overexpose the man and the trees while revealing noise in the darkest areas. ASTA removes the noise and then our tone mapper processes the result to reveal all of the original scene objects with reduced noise and improved saturation.

Figure 28: Two frames of SWIR video depicting walking in a forest scene. From left to right: the original frame, a histogram stretched version of that frame, the frame histogram stretched and processed with ASTA, and the frame processed using our full ASTA and tone mapping pipeline

In Figure 29 we present another SWIR video of a car driving at night without the use of headlights. The only well-illuminated pixels in the video come from engine heat being reflected by the ground under the car. As before, applying gain adds too much temporally varying noise and overexposes the video. ASTA reduces the noise and allows for a tone mapped version that gives a clear image of the car along with the surrounding terrain.

Figure 29: Two frames of SWIR video depicting a car with no headlights. From left to right: the original frame, a histogram stretched version of that frame, the frame histogram stretched and processed with ASTA, and the frame processed using our full ASTA and tone mapping pipeline

To illustrate that our system is flexible enough to work in a variety of environments, we used a similar setup as before, but this time the car headlights were on, thus overexposing the footage (Figure 30). Applying gain only makes the overexposure worse, but our tone mapping algorithm is able to salvage as much of the footage as possible and still able to enhance the dark areas.

Figure 30: Two frames of SWIR video that are overexposed due to a car with its headlights on. From left to right: the original frame, a histogram stretched version of that frame, and the frame processed using our full ASTA and tone mapping pipeline. Note how the tone mapping controls the blooming, allowing the car to be seen in addition to the surrounding scene.

When ASTA is looking at a static pixel, the result of that pixel in a processed video should approach a constant number, regardless of the intensity of the zero-mean noise. The top of Figure 31 shows a frequency histogram of luminance values captured at a single pixel of a static scene over 60 frames. Although the value should be 73, note the variance in the signal due to sensor noise. Our algorithm, as more frames becomes available to average, quickly converges to the correct result, a constant value of 73, as shown in the bottom histogram, verifying our noise assumptions.

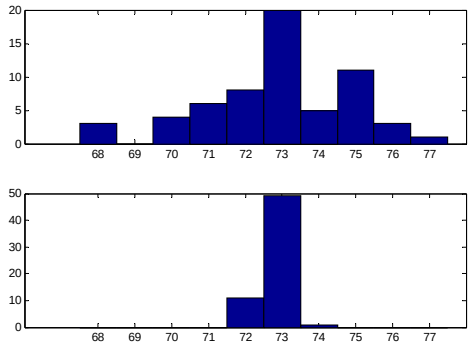


Figure 31: Two histograms of the luminance frequencies of a single static pixel through 61 SWIR video frames, before and after ASTA processing (no tone mapping). The original noisy signal (top) has a much greater variance than that of the ASTA processed result (bottom).

In Figure 32 and Figure 33, more ASTA and tone mapping results are shown, this time using color video RGB video captured with the Sony DFW-V500 imager. This time, pseudo-color images are shown to illustrate where ASTA chooses to use temporal integration (red) and where it chooses to use spatial integration (green). Note that many pixels are a blend of red and green, as some, but not all samples were obtained temporally, with the others coming from spatial neighbors.

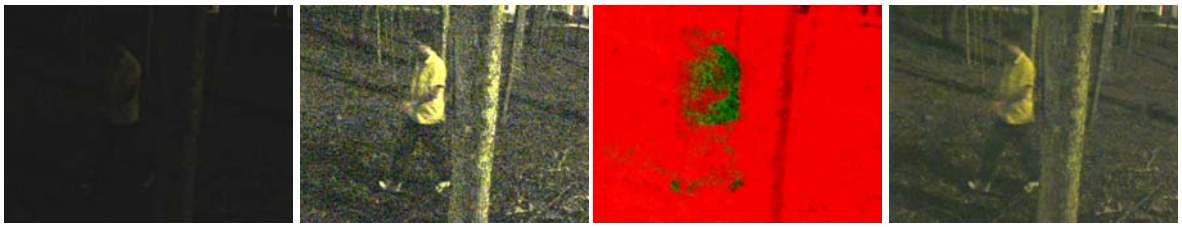


Figure 32: A frame from a video processed using ASTA and tone mapping. From left to right: Original frame, histogram stretched version, pseudocolor version (red = number of temporal pixels integrated, green = number of spatial pixels integrated), and our result after our full processing pipeline.

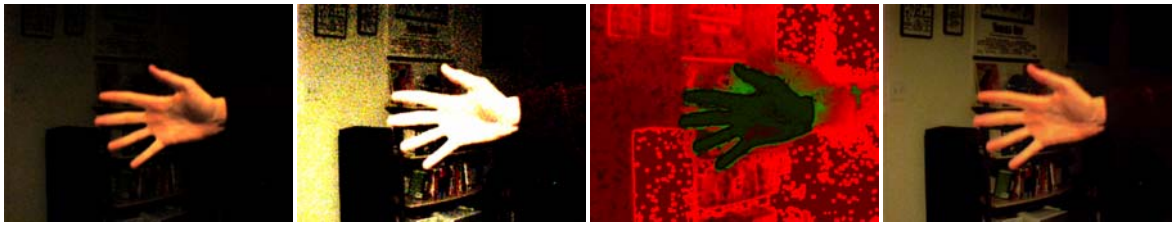


Figure 33: A frame from a video processed using ASTA and tone mapping. From left to right: Original frame, histogram stretched version, pseudocolor version (red = number of temporal pixels integrated, green = number of spatial pixels integrated), and our result after our full processing pipeline.

Figure 34 shows the benefit of our algorithm on image statistics, specifically in regards to luminance distributions. By starting with dark surveillance footage, it is easy to see how much quantization error appears when a gain is applied, which accentuates the noise. By working in floating point, ASTA can remove the quantization error, so that when the tone mapping expands the histogram, a much smoother luminance distribution is created.

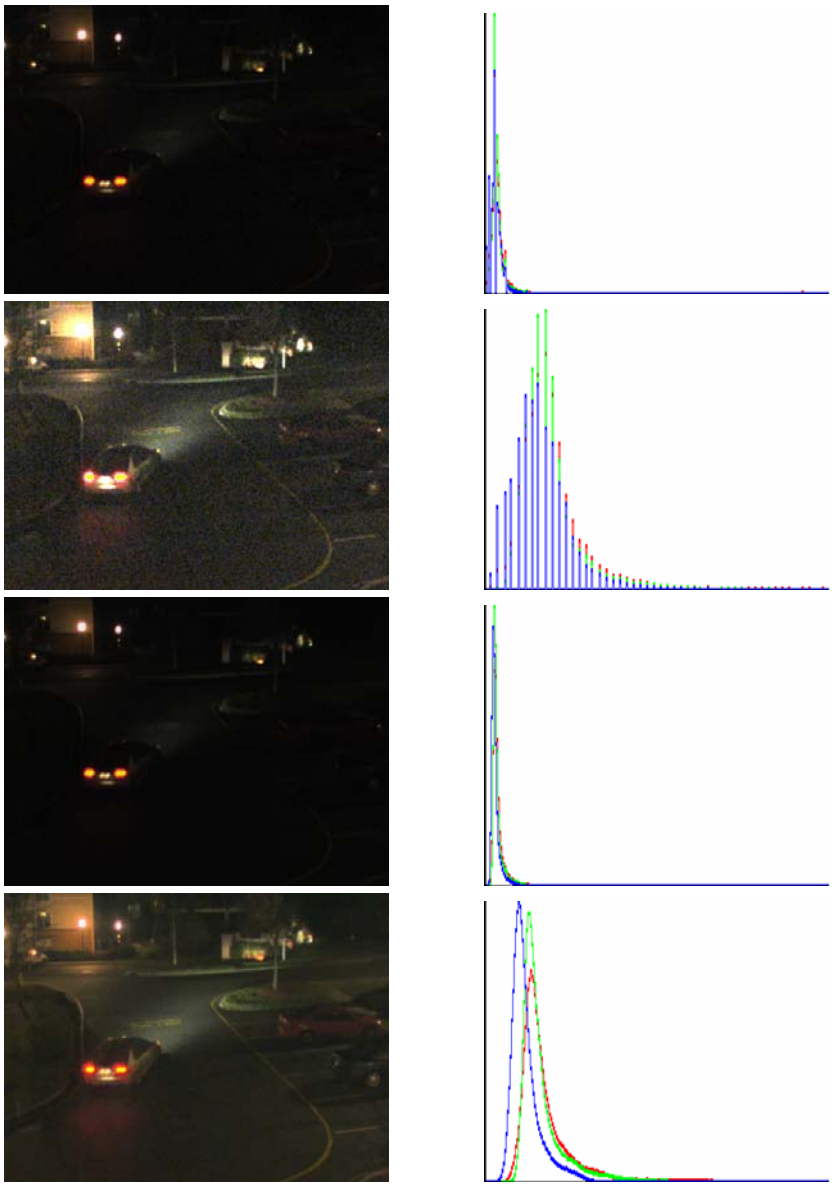


Figure 34: Inspection of color histograms in our process. From top to bottom: the original video frame and its histogram; a histogram stretched frame and its histogram showing quantization error; an ASTA processed frame and its histogram which is similar to the unfiltered histogram; the tone mapped ASTA frame and its stretched histogram without quantization error. Note the vertical scale in these histograms is vertically stretched to show maximum detail in each.

3.6.2 Colorization

In this section, we present the results colorization using each of the colorization algorithms discussed in Section 3.3.

3.6.2.1 Poisson Colorization

For this data set, the boundary conditions for Poisson interpolation are provided by a color picture taken from a RGB camera in a bright environment. The UV channels of the UV color space also come from this imager. The gradient field comes from gray images taken by the SWIR imager in a dark environment. Also, the RGB image and the gray image come from the same still scene captured with our calibrated beamsplit RGB and SWIR setup (Figure 35).

Figure 35: The registered RGB (left) and grayscale SWIR (right) input images

A simple approach is to simply copy the UV chrominance from the RGB image directly to the gray image (Figure 36). A better approach is to use Poisson integration on the luminance channel. The four edges of the RGB image are used as boundary conditions. UV chrominance is copied directly from the RGB image to the Poisson interpolated image. This creates somewhat better color saturation in the output image.

Figure 36: Results from chrominance transfer. Left: UV chrominance transfer from the RGB to the SWIR image. Center: SWIR image interpolated with the RGB boundary conditions. Right: Chrominance transfer from the RGB image to the interpolated image.

The next experiment highlights how this approach fails when there is a change in the scene composition. Again, the same approach as before is used, except the SWIR image contains an arm not present in the color scene. This results in incorrect colorization in all of the resulting images, shown in Figure 37**.**

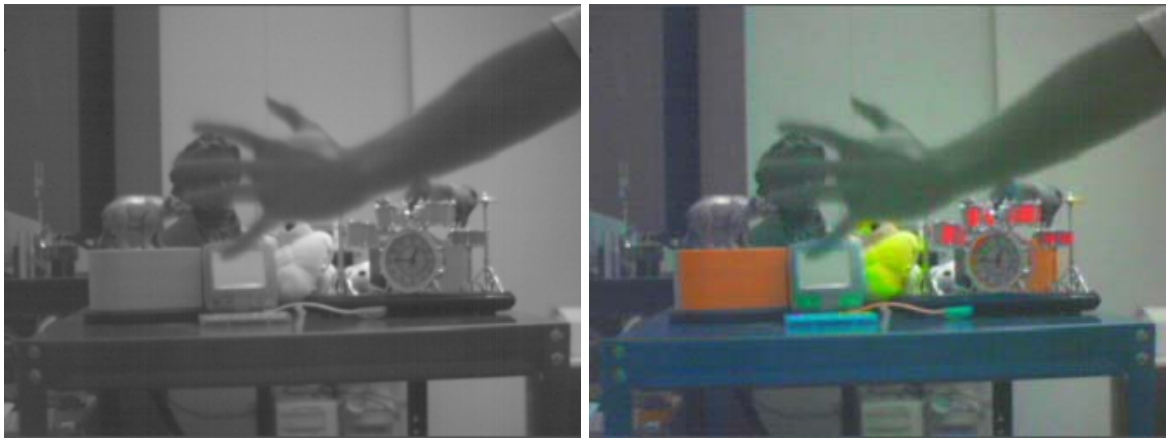


Figure 37: Results from a failed chrominance transfer. Left: SWIR image interpolated with the RGB boundary conditions. Right: Chrominance transfer from the RGB image to the interpolated image.

We can improve the results by only reintegrating the area around the arm, by using a mask with non-rectangular boundary conditions, shown in Figure 38.

Figure 38: Improved chrominance transfer using non-rectangular interpolation boundary conditions. Left: The mask defining the boundary condition. Center: The interpolated SWIR image. Right: Chrominance transfer from the RGB image to the interpolated image.

We now present a similar series of experiments where instead of grayscale SWIR images, we start with much darker grayscale images captured with a visible spectrum camera (Figure 39).

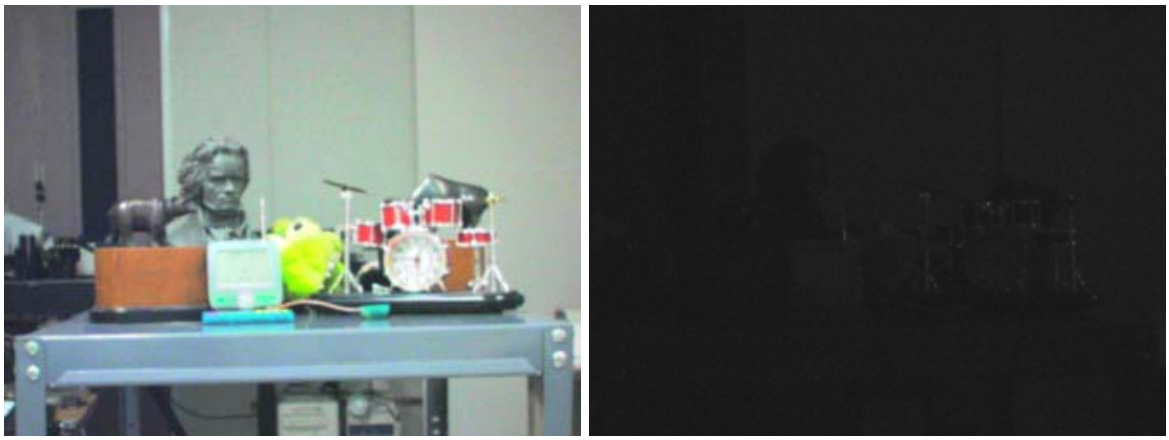


Figure 39: The registered well-exposed RGB (left) and under-exposed grayscale (right) inputs.

Again, we can either directly copy the UV color or re-interpolate using the well-exposed boundary conditions. Neither approach works particularly well, as shown in Figure 40.

Figure 40: Results from chrominance transfer. Left: UV chrominance transfer from the RGB to the dark image. Center: The dark image interpolated with the RGB boundary conditions. Right: Chrominance transfer from the RGB image to the interpolated image.

Interpolating with a gradient field containing a new scene element, as before, fails because the gradient field in the darker image is considerably darker than in the brighter image (Figure 41).

Figure 41: Chrominance transfer using non-rectangular interpolation boundary conditions. Left: The mask defining the boundary condition. Center: The interpolated SWIR image. Right: Chrominance transfer from the RGB image to the interpolated image.

Using a brighter version of the "dark" image creates better results. It is important to note that these images need to be relatively noise free to work, as noise makes Poisson interpolation create non-optimal results. Figure 42 shows results from this data set.

Figure 42: Results from a somewhat brighter "dark" image. Left: UV transfer to an interpolated image using rectangular boundary conditions. Center: Interpolation of an arm from the dark scene into the bright image. Right: UV transfer from the RGB image to the interpolated image.

Finally, we show these techniques using our forest environment. First, we show a simple color copy from a non-registered (but similar) daytime image to a SWIR image in Figure 43.

Figure 43: The non-registered RGB (left) and grayscale SWIR (right) forest images

Figure 44: Results from chrominance transfer. Left: UV chrominance transfer from the RGB to the SWIR image. Center: SWIR image interpolated with the RGB boundary conditions. Right: Chrominance transfer from the RGB image to the interpolated image.

Figure 45: Chrominance transfer to a video with a walking person. Left: The ASTA noise reduced video frame interpolated with the RGB edge boundary conditions. Right: Color transfer from the RGB daytime image to the interpolated image.

Figure 46: Improved chrominance transfer using non-rectangular interpolation boundary conditions. Left: The mask defining the boundary condition. Center: The interpolated SWIR image. Right: Chrominance transfer from the RGB image to the interpolated image.

Figure 44, Figure 45, and Figure 46 show a similar pipeline colorizing ASTA using a RGB daytime image for both boundary conditions and chrominance information. Note again that we have improved results when using the tighter interpolation bounds of the masked image.

Although Poisson techniques show promise for integrating together luminances from different sources and illuminations, the only reason chrominance can be copied is because there is a one-to-one correspondence between the source and destination pixels. As shown above, as soon as a new object enters the scene, the chrominance is wrong. For this reason, we consider machine learning approaches that can colorize a scene based on a known image prior.

3.6.2.2 Learning-Based Colorization

Unlike the previous section, where the color and gray images needed to be well registered to directly copy chrominance, learning-based colorization is more flexible and has more potential applications. Now, our only requirement is that the target gray image is similar to the source color image to some extent. The algorithms find correlations between the target gray image and source color images based on some constraints and then colorize the gray image. In this section, we will show the colorization results obtained by an existing mean and variance based algorithm [Toet 05] and our new belief propagation based algorithm.

We begin by looking at a simple example in Figure 47. The source and target images were taken in the same scene with some camera movement. The gray image is the extracted luminance channel from the color image. Both the mean and variance based algorithm and our belief propagation-based algorithm were used to colorize the gray image. In this example, the known ground truth can help us evaluate the success of our method. We can notice that some parts of the colorization result obtained by the mean and variance based algorithm are not correct, such as the roof of the structure, where the red colors contaminate the white and blue parts. For the belief propagation-based algorithm, the chrominance segmentation is shown along with the colorized result. Because this algorithm segments the chrominance well, the learned chrominance regions can be correctly copied to the target image.

Figure 47: A simple example of learning-based colorization. The top left is the source color image. The top middle is the ground truth. The top right is the synthetic gray image obtained by extracting the luminance channel from the ground truth. The bottom left is the colorized result based on mean and variance [Toet 05]. The bottom middle is the chrominance segmentation of the source color image. The bottom right is the colorized result based on belief propagation.

The second example, shown in Figure 48, uses images taken outside an apartment complex. The color image was taken during the daytime and the gray image was taken during the nighttime. We observe that the colorization result obtained using the mean and variance algorithm is very noisy. Some red patches appear on the grass and reds and grays mix together on the building's brick façade. Our belief propagation approach gives a less noisy result. The grass is evenly green and the building is represented by only one color. However, the color of the wall is not red. This is because the intensities of the wall and the roof are similar in the gray image and lack sufficient texture information. Therefore, they cannot be distinguished by low-level constraints.

Figure 48: Colorization of an apartment scene. The top left is the source color image captured during the daytime. The top middle is the gray image captured in night time. The top right is a histogram stretched version of the gray image. The bottom left is the colorized result based on mean and variance. The bottom right is the colorized result based on belief propagation.

Finally, our third example shows color transfer from an RGB image to an IR image of an indoor scene. The color image was taken from a RGB camera in a bright environment and the grayscale image was taken with a SWIR imager in a dark environment. Both images were taken using the aforementioned beamsplitter setup, allowing the cameras to share an optical path. In this example, the luminances of the color image and the SWIR significantly differ and there is insufficient texture information for classification. Chrominance transfer fails because it is not feasible to directly transfer chrominance from the color image to grayscale SWIR image with so little low-level texture distinction. Regardless, we can see the belief propagation-based colorization still outperforms the mean and variance-based algorithm.

Figure 49: Colorized SWIR image based on a pair of examples. The top left and top middle are the source image pair of a registered RGB image and a grayscale SWIR image. The top right is the target SWIR image. The bottom left is the colorized result based on mean and variance. The bottom right is the colorized result using our belief propagation-based technique.

3.6.3 Luminance Transfer

The relative luminance intensities of the visible spectrum do not always correspond to the relative luminance intensities in the IR domain. Subsequently, just performing chrominance transfer is insufficient to give IR images the appearance of daytime example RGB images. Like color transfer, the luminance of the IR image can be adjusted based on example images. We show that colorizing luminance adjusted IR images gives visually superior results.

Because of the imager availability constraints, we used Near IR (NIR) images instead of SWIR images for these experiments. A visible spectrum filter is put in front of a CMOS camera sensitive to the NIR spectrum to capture NIR images. In all experiments, we learn from a pair of registered NIR and RGB images. For the target NIR image, the luminance is adjusted based on the source images. We then colorize the luminance adjusted images using the previously discussed algorithm.

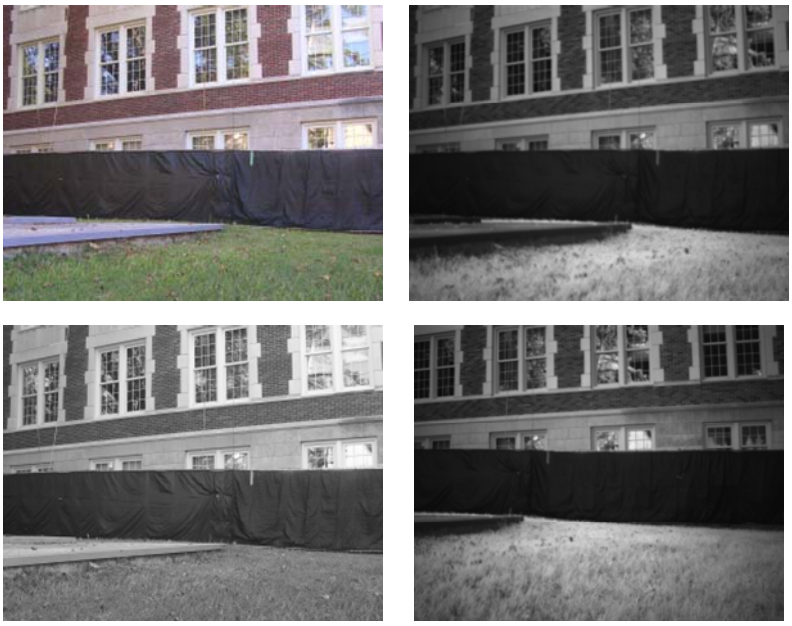


Figure 50: The inputs to our first example. The top left is the source RGB image. The top right is the source NIR image. The bottom left is the luminance channel of the source RGB image. The bottom right shows the target NIR image.

The inputs to our first example are shown in Figure 50. The RGB and NIR images are both taken during the daytime. Camera movement is introduced between the source images to the target image. Because the images are captured in different spectra, the relative luminances in the RGB image and the NIR image differ. Comparing the NIR image to the luminance channel of the RGB image, the grass is brighter and the fence and building are darker.

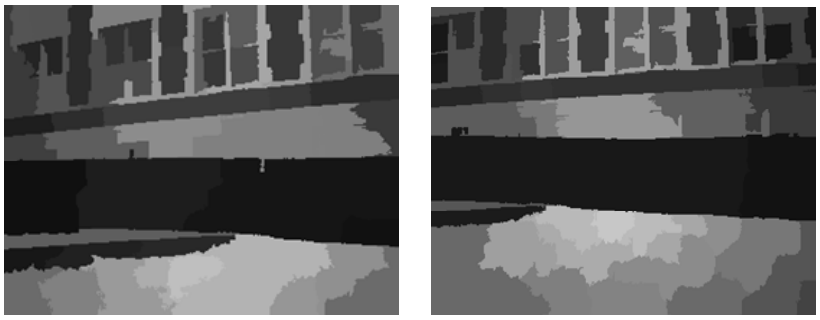


Figure 51: The segmentation of the near IR images. The left is the segmentation of the source near IR image. The right is the segmentation of the target near IR image.

The starting point of the luminance transfer is to segment both the source NIR image and target NIR image. Figure 51 shows the results of mean shift segmentation using the implementation of Comaniciu et al [02] configured to over-segment the images. In this experiment, there are 51 and 54 regions in the source and target images, respectively.

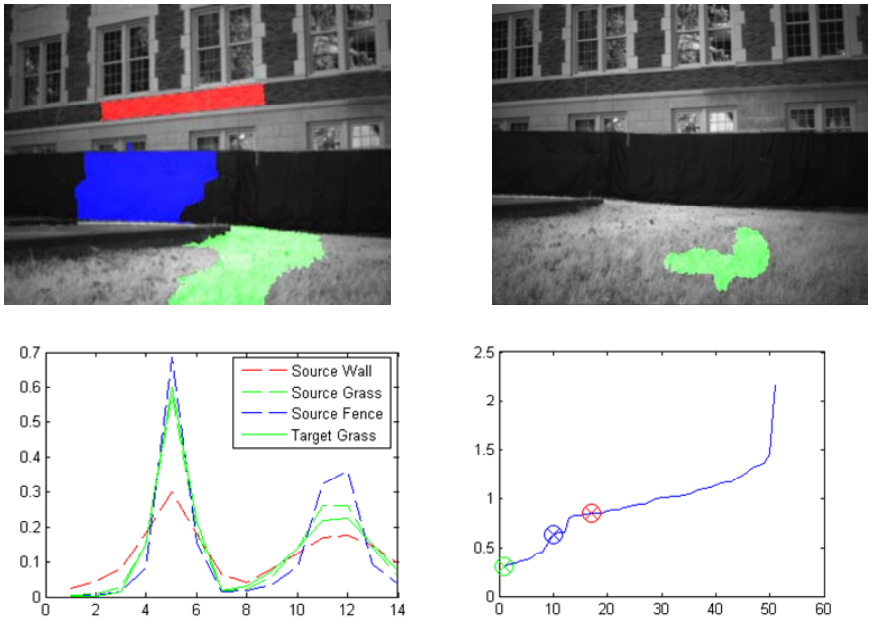


Figure 52: The result of feature vector extraction and matching. The top left is the source NIR image with three selected regions. The green region grass, the blue region is fencing, and the red part is brick. Top right is the target IR image with a selected grass region. The bottom left is the feature vectors of the selected regions. The total length of each feature vector is 300. Here we only show a segment of them. The dashed lines are feature vectors of the selected regions in the source image with corresponding color. The solid line is feature vector of the selected region in the target image. The bottom right is the sorted similarity of the selected region in the target image and all regions in the source image. The green, blue and red marks represent the selected three regions in the source image.

Figure 52 shows the result of feature vector extraction and matching. 5 levels of wavelet decomposition are applied. Each level has three coefficient arrays. We use 20 bins for each histogram. The length of each feature vectors is therefore 300 coefficients. These settings allow the feature vectors to classify the texture properties of each region sufficiently well. From the plots of the feature vectors and similarity measures, we can see the selected grass region in the target image is most similar to the grass region in the source image.

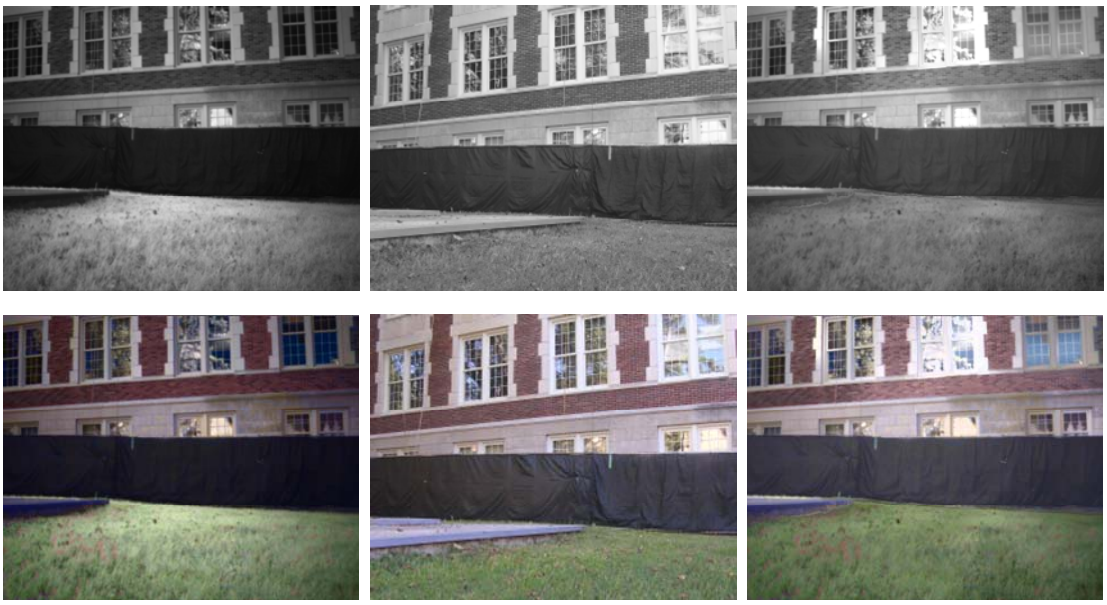


Figure 53: Luminance and color transfer results. The top left is the target NIR image. The top middle is the luminance channel of the source RGB image. The top right is the luminance adjusted target NIR image. The bottom left is the colorized target image without luminance transfer. The bottom middle is the source RGB image. The bottom right is the colorized target image with luminance transfer.

Luminance and color transfer results are shown in Figure 53. For comparison, we also show the source RGB image and the colorized target images without luminance transfer. After transfer, the grass becomes darker and the fence and building become brighter. Compared to the colorization without luminance transfer, the colorized target image with luminance transfer is more similar to the source RGB image and has a more natural appearance.

Figure 54 shows another outdoor example. The source color image was taken in the daytime with an RGB camera. The NIR images were taken at night with a long exposure. Again, there is camera movement from the source images to the target image. The luminance of the sky in the NIR images is saturated because of the long exposure. Thus, directly colorizing the target NIR image causes unnatural coloration of the sky. Luminance transfer successfully makes the sky darker. Again, the overall appearance of the colorized target image with luminance transfer is more similar to the source RGB image.

Figure 55 shows a more difficult example. In this example, the source images and target image are taken from different scenes instead of same scene with camera movement. Luminance transfer improves the appearance of the target NIR image. After learning the relative luminances from the source RGB image, it makes the building, the ground, and the sky brighter while making the tree darker in the output. More details from the ground and the building can therefore be seen and the image generally contains more daytime visual characteristics.

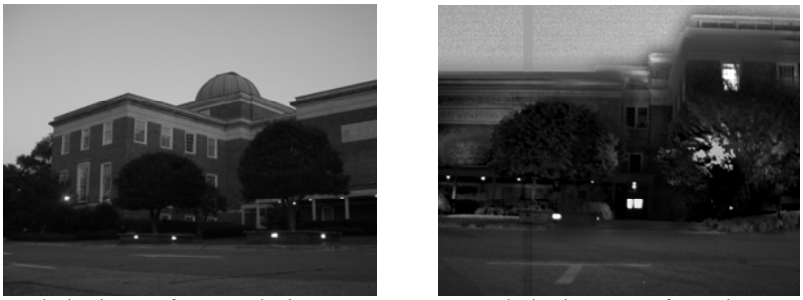
The source color image | The source near IR image | The target near IR image

The luminance of source color image | The luminance adjusted result

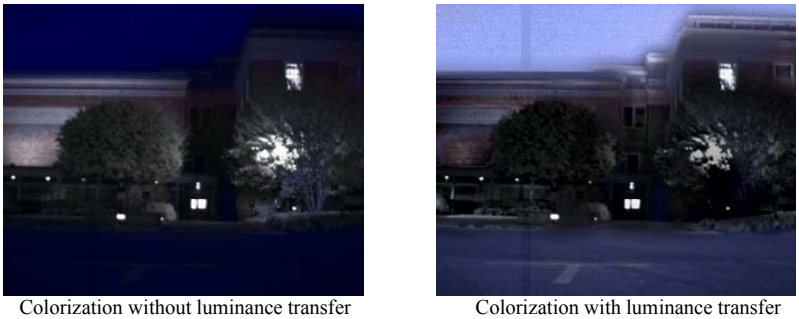
Colorization without luminance transfer | Colorization with luminance transfer

Figure 54: The second example of luminance and color transfer.

The source color image | The source near IR image | The target near IR image



The luminance of source color image | The luminance transfer result



Colorization without luminance transfer | Colorization with luminance transfer

Figure 55: The third example of luminance and color transfer.

3.6.4 Multi-Spectral Bilateral Fusion

To present our results from the multi-spectral bilateral fusion technique, we show corresponding frames from our RGB and SWIR cameras captured using a cold mirror beamsplitter. All frames shown here have been pre-registered to provide both horizontal flipping (to remove the mirror reflection inversion) and fine alignment.

Figure 56 shows the before and after of a frame from a video. This is a straightforward example because there is no motion, and thus no motion blur in the RGB footage to combat. Figure 57, however, shows a moving hand which is blurred in the RGB footage due to motion combined with a longer exposure time. The multi-spectral bilateral fusion cleans up this blurring with the joint bilateral filtering. Finally, Figure 58 shows how we can use this technique to restore RGB facial features which are much clearer and recognizable in the SWIR footage. Despite the noisy RGB, we are still able to extract usable chrominance for use in the final result and simultaneously maintain proper relative luminance levels in all of our examples.

Figure 56: Inputs and output of our multi-spectral bilateral fusion technique on a video frame. Upper Left: Original RGB frame; Upper Right: Histogram stretched original RGB frame; Lower Left: Corresponding SWIR frame; Lower Right: Multi-spectral bilateral fusion result.

Figure 57: Inputs and output of our multi-spectral bilateral fusion technique on a video frame. Note, the output video contains the reconstructed fingers without the motion blur in the original RGB footage. Upper Left: Original RGB frame; Upper Right: Histogram stretched original RGB frame; Lower Left: Corresponding SWIR frame; Lower Right: Multi-spectral bilateral fusion result.

Figure 58: Inputs and output of our multi-spectral bilateral fusion technique on a video frame. The facial features and clothing details are effectively copied from the SWIR footage into the color result. Upper Left: Original RGB frame; Upper Right: Histogram stretched original RGB frame; Lower Left: Corresponding SWIR frame; Lower Right: Multi-spectral bilateral fusion result.

DISCUSSION

In this section, we discuss our accomplishments in terms of each of the originally proposed objectives. In the agreement, the original Statement of Work overview for all 4 project phases reads:

"The UNC team will investigate the feasibility of applying Poisson interpolation to the problem of enhancing night-vision(NV) images and evaluate the viability applying machine-learning approaches to the problems of (a) estimating daytime reflectance from local texture information and (b) combining gradient images from multiple NV sensors to estimate daytime gradients.

The UNC team will initiate development of fast and efficient algorithms for Poisson interpolation, multispectral (MS) recognition, and classification with the goal of enabling current multispectral NV systems or new sensor systems to optimize MS-NV image reconstruction to achieve daytime image quality.

The UNC team will plan work on next generation NV capabilities which will provide imagery that will in many ways surpass daylight images in both information content and quality.

Application of interest for this study include wearable visualization electronics for pilots and dismounted tactics operators"

This document discussed the first year (Phase I) of the agreement in further detail:

"In this first phase we will investigate the feasibility of applying Poisson interpolation to the problem of enhancing night-vision images. We will also evaluate the viability of applying machine-learning approaches to the problems of 1) estimating daytime reflectance from local texture information seen under night vision conditions and 2) combining gradient images from multiple night-vision sensors to estimate daytime gradients. A central challenge of this approach will be in determining how well accurate gradient data can be extracted from night-vision images. Poisson integration methods are sensitive gradient accuracy and gradients themselves tend to be sensitive to noise. We contend that much significant gradient information is invariant to wavelength, in particular, object and material boundaries. A large component of this phase of research will be the development and evaluation of algorithms and defining the principles of their operation. The prototype systems developed in this phase will most likely be built upon rapid prototyping systems and languages such as Matlab. We do not expect to achieve performance levels that are near real-time."

For Phase I, the deliverable was a white paper detailing the components and algorithms used to achieve our goals. This document constitutes that white paper.

To address the proposed goals, we break the problem down into four areas that encapsulate all of the original project aims: noise reduction, tone mapping/contrast enhancement, colorization, and luminance transfer.

4.1 Noise Reduction

To evaluate our technology, we will compare our capabilities to those of other available applications for similar reduction of temporal noise. The noise reduction available in mainstream professional video-editing applications, such as Apple's Final Cut Pro, can remove noise via temporal averaging, but results in ghosting for both moving camera and moving object situations. Due to the number of frames necessary to average to effectively remove noise in LDR video, this ghosting can become severe. The software provided by Sensors Unlimited with their SWIR imager can run in high speed mode and combine several sub-exposures into a single exposure. However, this again can cause ghosting of moving objects and scenes, because it assumes very little motion between exposures.

Our ASTA noise reduction, described in Section 3.1.3, has been shown to effectively remove noise from a variety of source footage types. We can handle static scenes with static cameras with similar levels of noise reduction as the above applications. However, we additionally handle scenes with moving objects and also non-stationary cameras. This allows our technology to have a wider range of applications, ranging from surveillance to night-vision goggle systems.

One of the key strengths of our approach to removing noise is that the same technique is effective for both visible and non-visible NIR and SWIR spectrum cameras (and their associated noise sources), as demonstrated in Section 3.6.1. This is because the current generation of sensors in all of these categories suffer from similar Gaussian zero-mean temporal noise. Also, because our system supports both uni- and multi-spectral imagers, such as grayscale and RGB, we can focus on optimizing and improving one single algorithm instead of a whole family of algorithms.

We realize that due to our reliance on non-linear filtering, the ASTA noise algorithm is not fast-enough to run in real time. As part of the agreement, we discussed creating Matlab implementations of our algorithms. Because ASTA was complex, it required a higher-speed C/C++ implementation in order to run at a rate plausible for processing hundreds of video frames. This version can process RGB and SWIR video at roughly one frame per minute. We anticipated that in the first phase, the algorithms would not run in real time. However, we have recently achieved an order of magnitude increase via fASTA (Section 6.2) and anticipate similar gains on existing hardware in the next phase using a variety of techniques (Section 6.3).

4.2 Tone Mapping / Contrast Enhancement

In this section, we evaluate how well we can enhance the contrast of an under-exposed image or video to resemble well-exposed daytime imagery or compress HDR imagery. We developed both iterative (Poisson) and non-iterative (bilateral) techniques to solve this problem.

We begin by discussing our non-iterative approach. We are faced with two problems in benchmarking. First, there are no "off-the-shelf" tone mapping applications on the market to compare ourselves against. Secondly, we are the first to address the issue of reversing the tone

mapping process to expand dynamic range, instead of the traditional approach of compressing dynamic range. Thus, we will compare our techniques based on methodology.

We consider the [Durand and Dorsey 02] tone mapping approach to be representative of a general, all-purpose tone mapper that behaves well in a variety of environments. Because it is designed to compress dynamic range, we reverse the algorithm to expand dynamic range. Note that this algorithm was not designed to be reversed, but it is the best alternative available. When doing so, it has a tendency to cause the largest relative increase in luminance in the darkest of pixels. This causes two problems to occur. First, due to the mapping curve, far too much of the dynamic range is wasted on a very small portion of the input luminances. Secondly, the darkest pixels are generally the least reliable indicators of high-frequencies, because we had to filter them very heavily to remove noise with ASTA.

Our technique solves this problem by expanding on the general approach of [Durand and Dorsey 02] by limiting the amount of high-frequency texture in the output based on how much filtering was necessary to clean a pixel value. This acts as a second noise reducing step and avoids accentuating any errors that survived the first tone mapping pass. We also adjust the mapping curve to better distribute input and output luminances between 0-255. The results of this tone mapping are shown in Section 3.6.1 in conjunction with ASTA. Because our tone mapping is superior when dealing with temporal noise, these still image results do not show the full extent of our improvement.

Our Poisson based approach results are shown along with the colorization results in Section 3.6.2.1. Our evaluation of the Poisson interpolation has led to the following conclusions. First, if a daytime image taken with the exact same camera parameters (intrinsic and extrinsic) as a nighttime image, interpolation of the nighttime image with the daytime boundary conditions works as expected. The images luminances are increased in an appropriate manner and the result looks plausible. However, as soon as the camera moves or a new object enters the scene, the results can become poor and difficult to interpret.

We found Poisson interpolation is often more successful when a mask is specified that indicates how to mix two gradient fields together. Although amplification of one of the fields may be necessary as a pre-process, this creates better results than simply using the rectangular boundary conditions and allows images to be fused together in a more interpretable manner. We have also discovered that this process is very sensitive to noise, requiring aggressive ASTA filtering in order obtain usable night-vision gradient fields.

After our analysis, we consider Poisson interpolation to be a useful tool for processing night-vision imagery to resemble daytime imagery, particularly for image fusion. However, it is but one of a suite of tools, along with machine-learning and non-linear filtering, that comprise our pipeline for video processing.

4.3 Colorization

We have presented two novel colorizations and analyzed the current state-of-the-art technique by Toet [2005]. Our first method consists of directly copying color from daytime RGB images to grayscale night-vision images taken with light-amplification and SWIR imagers. We tone map these using new Poisson boundary conditions and directly transfer chrominance. In this scenario, the result is acceptable, but as soon as the camera or scene moves, the luminances and chrominances no longer match and the output is unacceptable.

For this reason, we now use machine-learning approaches to transfer color between areas of similar visual characteristics. Our belief-propagation approach, which uses a variety of image comparators, including filter banks, creates results that are comparable and often better than the Toet [2005] method. These results, shown in Section 3.6.2.2, show that by using more robust classifiers, we can properly handle colorization of highly textured regions. For this reason, we believe we have the most general single-band image colorization algorithm.

We do recognize that a trait of all current algorithms is that high quality colorization of non-visible-band imagery requires proper relative luminances. If a pixel is incorrectly displayed as black, no amount of colorization will change the color of a black pixel. Basically, we have seen in many circumstances, especially with SWIR, that without proper relative luminances, the result is still sub-optimal. Thus, we feel that colorization and luminance transfer algorithms are both necessary to create acceptable output.

4.4 Luminance Transfer

The process we dub luminance transfer, the processing of non-visible spectra imagery to have the same relative luminances as daytime imagery, does not have any significant prior research, so we describe our progress in the area.

We have discovered that although the problem is related to chrominance transfer (colorization) (to use the more common nomenclature), the problem is far more difficult. This is because errors are much more noticeable due to the human visual system's acuity for edge detection and classification. Subtle colorization errors, on the other hand, are often acceptable, which is why color is undersampled in many sources, such as broadcast television.

Furthermore, the problem appears to be less constrained than colorization because there does not appear to be as nice of a one-to-one correspondence between texture and proper luminance. Whereas we have been able to classify regions of color by texture encoded in luminance, there is less of a relationship between average luminance levels and the texture encoded in the luminance.

However, we believe the algorithms shown in Section 3.4 and the results in Section 3.6.3 indicate that further research is warranted.

4. CONCLUSIONS

In this research program, we have developed and assessed a suite of new tools for enhancing night-vision imagery. The primary innovations of our approach are the use of a virtual imaging array where every photosite has its own independent exposure time, and the application of Poisson integration to transfer color and luminance information while preserving local details. The largest component of this phase of our research was the development and evaluation of algorithms and defining the principles of their operation.

We implemented a virtual per pixel exposure by maintaining a delay line of video frames in a spatio-temporal volume. This enables a virtual exposure to be set according to the ambient light level falling onto each pixel, and its nearby neighbors. The exposure level, in our approach adjusts adaptively to obtain the desired signal-to-noise ratio and image contrast. This contrast adjustment considers both the temporal and spatial neighborhoods and noise levels of each pixel, or photosite. This allows the illumination-to-luminance mapping function to adapt dynamically over the entire image, thus enhancing contrast and brightness levels, while maintaining a perceptually faithful image.

Our Poisson integration approach enables us to incorporate perceptually important details that are either not present (but predictable) or not well sampled in the low-light images. For example, sparsely sample color data can be integrated across the entire scene, dramatically improving the comprehensibility. We have shown that Poisson integration can also be used to fuse multi-spectral images. In particular, we have shown how the high quality gradients visible in a SWIR image can be integrated with low-light visible images as boundary conditions to provide high-quality images that appear to be more natural than the original SWIR images. A central challenge of our research has been in determining how well accurate gradient data could be extracted from night-vision images. Poisson integration methods are sensitive gradient accuracy and gradients themselves tend to be sensitive to noise. We have demonstrated with our transfer of SWIR gradient information using visible light boundary conditions our contention that much significant gradient information is invariant to wavelength. This enhancement will improve comprehension and it may reduce training time and user fatigue.

The local contrast adaptation capability of Poisson integration also has the potential to make solid-state night-vision systems immune to night-vision countermeasures. Poisson integration allows a high-dynamic range images to be accurately depicted. This means that images with both bright and low-light pixels, varying by as much as four orders of magnitude, can simultaneously imaged and displayed, even on low dynamic range displays. This contrasts with the capabilities of standard cameras that tend to either locally bloom (become fully bright), or globally saturate (loosing details in dark regions). The combination of our tone-mapping approach with Poisson integration provides this capability.

We also able to remap unlit visible images and images in non-visible parts of the spectrum to simulate a more interpretable illumination condition. This is accomplished using machine learning methods. We are able to segment and classify a training set of images and transfer their qualitative properties to other images. For instance, we can map the unnatural response of SWIR

imagery so that it has an appearance more similar to a visible-light luminance image. A variant of the same method permits us to map low-light images to approximate the appearance of daylight images. We accomplish this using an example-based classifier to learn the mapping. This technique assumes that the training dataset is representative of the image whose response is to be relit. These methods can also be combined with the Poisson integration method to improve contrast and apply colorization.

Currently, all of the methods are computationally intensive and ranges from minutes to a few seconds pre frame processed. Our ultimate goal is to provide similar capabilities in real-time as mentioned in the Discussion section. The second phase of our research will focus on the development of fast and efficient algorithms for noise reduction, tone-mapping and Poisson interpolation. We will also enhance the backend of the multispectral classification and remapping process. In this phase of the research, we request a grid-based multiprocessor for developing and simulating fast distributed processing algorithms for Poisson equation solving and for analyzing daytime training data sets. At the end of this phase we expect to have a working interactive prototype for evaluation, and a better understanding of the computational limits of our approach to night-image enhancement.

In the third phase of our research, we hope to be teamed with providers of existing multispectral night-vision systems or creators of new sensor systems. We have made initial contacts with Sensors Unlimited Inc, and they have been helpful in loaning us a SWIR camera for evaluation of our imaging techniques. We plan to continue this collaboration and provide them with our registered video sequences in both raw and processed form for evaluation.

5. RECCOMENDATIONS

In this section, we detail research directions that address both the goals originally outlined in the project proposal as well as pertinent new directions related to discoveries made in the first phase.

6.1 Wavelet Based Multi-Exposure Integration

The results of ASTA have shown that effective noise reduction can be performed by combining the luminance contributions of multiple frames together followed by performing contrast enhancement. Depending on the noise amplitude, a large number of samples could be needed to average out the sensor's Gaussian noise. In situations where fewer samples are available than necessary, non-moving areas can exhibit spatially-static, slowly-varying temporal noise. In order to remove this distraction and decrease the number of samples needed in active core system memory for all processing, we have begun to explore more robust multi-frame luminance integration methods.

A promising option is to use wavelet-domain techniques to replace the arithmetic-mean currently in use. Wavelet techniques have been successfully applied to noise reduction and image compression problems. However, we are interested in how best to use them to coherently process noisy video.

Wavelet decomposition operates within an image pyramid, where each higher level has half the spatial resolution of its lower neighbor. The decomposition gives an amplitude for each of the wavelet basis functions at each pyramid level. Noise reduction has been performed by previous researchers by removing all contributions of wavelet basis functions whose amplitude is below some given threshold [Fodor and Kamath 03]. When the image is reconstructed from the new wavelet basis function histograms, noise is removed. Doing the processing using wavelets as compared to a spectral density approach (e.g. in the Fourier domain) gives far more local spatial control, as the wavelet compositions can be varied at any scale and at any location.

We intend to extend this approach to utilize the wavelet decompositions of multiple frames simultaneously to make more robust choices about which basis functions to eliminate. Specifically, this approach would not average frames together directly, but instead would average together their wavelet-domain decomposition histograms.

Making this technique effective involves research into a number of areas. Primarily, an operator other than simple thresholding must be used in order to be robust to important subtle details while removing noise. Such an operator could be per-pixel, or reach consensus with spatial or temporal neighbors via a belief propagation approach [Malfait and Roose 97]. Consistency is important to guarantee uniform decisions at all pyramid levels.

Another area of research is how to integrate this system with ASTA to take advantage of ASTA's ability to properly handle areas of motion and non-motion. Making sure that areas of motion do not contaminate the statistics of static areas could be accomplished through ASTA's motion detection and weighting scheme.

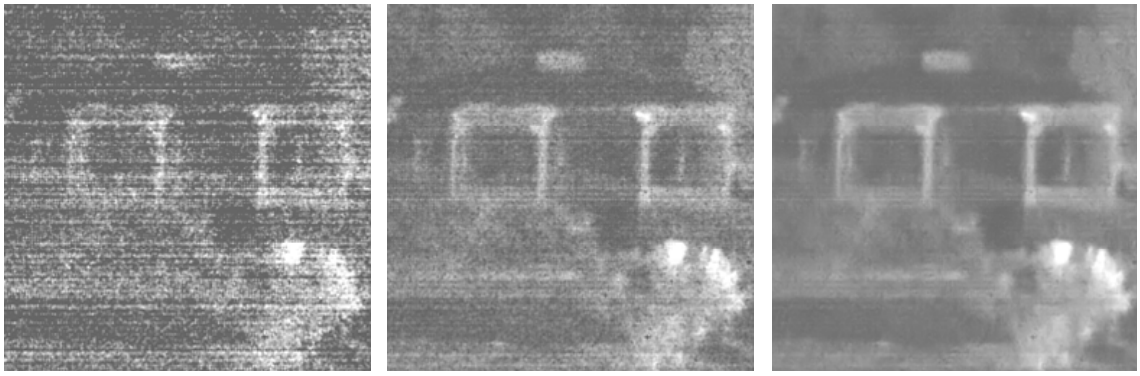


Figure 59: Early results of the wavelet multi-exposure integration technique. On the left is one of the 11 input frames of a static camera sequence. In the middle is the arithmetic mean of those images. On the right is the wavelet-based technique.

Figure 60: This shows early results of the wavelet noise reduction on 11 frames of the supplied SWIR video in a static area of the video. On the left is one of the input frames of a static camera sequence. In the middle is the arithmetic mean of those images. On the right is the wavelet-based technique.

6.2 Performance Enhancement

The FilterShop application now includes a function entitled fASTA, or fast ASTA. This is an early implementation of an accelerated version of the ASTA noise reduction and tone mapping algorithms. fASTA contains similar functionality to the original algorithms, but is designed with greater memory coherence, especially in the management of the cache of frames being combined and denoised (which can be on the order of hundreds of megabytes or gigabytes).
The only functionality removed for the sake of speed was the temporal smoothing used in tone-mapping, which was very slow and was responsible for only a small increase in quality. Also, it improves speed by avoiding many slow edge condition calculations by removing the outer border of 10 pixels, a common technique in numerical methods.

The published version of the ASTA and tone-mapping algorithms often took on the order of two or more minutes to robustly process each video frame. The current version of fASTA can process the same high-resolution video in less than 20 seconds per frame, and some lower-resolution examples at roughly 3 seconds per frame. However, two more orders of magnitude speed improvement are still needed to reach real time. Continuing work will be in tweaking these algorithms and finding functionally similar, yet more efficient alternatives. Please refer to the FilterShop manual for instructions and specifications.

6.3 GPU / Multi-Processor / Multi-Core Implementation

In order to achieve real-time performance, it will become necessary to pursue computation outside of the single CPU model. Offloading some of the highly parallelizable aspects of the application to the Graphics Processing Unit (GPU) is possible, considering that the necessary floating-point precision is now found on high-end graphics cards. Because this will not likely alone yield real-time performance, the work must be distributed over multiple processors, either in shared-memory multi-core machines, or over high-speed distributed processing clusters.

The early challenges of distribution will involve scheduling and memory/cache allocation issues. Because ASTA does not spend an equal amount of time processing each pixel (darker pixels must be robustly handled, and moving pixels require additional spatial filtering) it is difficult to predict exactly how to distribute the workload a priori. When work is redistributed on-the-fly, the additional processing units must access the pertinent areas of the spatio-temporal video volume, which requires loading large chunks of data from a remote computer or disk. Therefore, careful caching is necessary to avoid incurring cache-miss penalties, which is crucial in a real-time environment.

Therefore, we have begun discussion with the real-time scheduling community at UNC Chapel Hill, whose current research involves a class of scheduling algorithms called PFair [Srinivasan and Anderson 03]. This system has the capability to dynamically re-weight processor share while still maintaining deadlines necessary to guarantee 30 frames per second video output. Furthermore, their group's recent research has been concerned with efficient caching of large data sets.

We will investigate the applicability of this scheduler and supporting technologies as a possible method to efficiently distribute our algorithms. An initial study of how the ASTA algorithm specifically could be used in a multi-processor environment has already occurred and is discussed in "Accuracy versus Migration Overheads in Multiple Reweighting Algorithms," by Aaron Block and Jim Anderson, which is currently under review at the 12th IEEE Real-Time and Embedded Technology and Applications Symposium.

6.4 Multi-Spectral Resolution/Frame Rate Enhancement

Fusion of information from multiple visual sensors of different spectra is becoming crucial in the modern battlefield. However, most fusion comes from basically registering the signals from

multiple sensors and then overlaying them. We wish to expand this notion to have more complex interactions between spectra. Possibilities for interactions include:

- Integrating multiple varying exposure rates and spectra into a single, high frame rate video by enhancing and de-blurring long exposure spectra with information from short exposure spectra.
- Improving noise vs. motion classification in visible spectra by analyzing non-visible spectra.
- Performing super-resolution enhancement of low-resolution spectra with information from high-resolution spectra.

6.5 Improved LDR Tracking Algorithms

We have demonstrated the effectiveness of our methods for moving cameras, but its success depends on the ability to reliably track features in an underexposed video sequence. This becomes more difficult when the image is composed of many independently moving regions. Our moving camera techniques currently stabilize only the single largest flow field, which was the background in our experiments. A more general solution would establish temporal correspondence for all image regions, perhaps by using optical flow methods. However, it is likely that typical optical flow techniques, which depend on robust gradient estimates, would fail on our noisy underexposed source images. Therefore, creating more robust algorithms would be necessary.

6.6 Extending Poisson Interpolation

In Section 3.3.1 we discussed the uses of Poisson interpolation for image processing, specifically to combine image information from both daytime and nighttime settings. Poisson interpolation is typically solved using iterative processes, but much acceleration is possible by using multi-grid solvers. However, enforcing boundary conditions on a per-pixel basis within existing multi-grid methods is an ongoing issue, particularly as it relates to guaranteed system stability and coherence over multiple video frames.

We are also continuing to evaluate the possibility that contrast fields, as opposed to gradient fields, may have interpolation properties that result in more desirable output. Using the (B-A)/(B+A) contrast definition, it is possible to evaluate Poisson equations within a more perceptually valid basis than simply log domain Poisson processing. However, in order to be viable, this solution needs to have guaranteed convergence, and also a stable multi-grid solver to improve interpolation performance.

6. SYMBOLS, ABBREVIATIONS, AND ACRONYMS

ACM	Association for Computing Machinery
AFRL	Air Force Research Laboratory
ASTA	Adaptive Spatio-Temporal Accumulation
C2ISR	Command, Control, Intelligence, Surveillance, Reconnaissance
CCD	Charge-Coupled Device
CMOS	Complementary Metal Oxide Semiconductor
DARPA	Defense Advanced Research Projects Agency
EM	Expectation-Maximization
fps	frames per second
HF	High Frequency
IR	Infrared (0.6-14 μm)
MAP	Maximum a Priori
MS	Multispectral
ML	Machine Learning
NIR	Near Infrared (0.6-1 μm)
NV	Night-Vision
RGB	Red Green Blue
SIGGRAPH	Special Interest Group on Computer Graphics and Interactive Techniques
SNR	Signal-to-Noise Ratio
SOF	Special Operations Force
SWIR	Short Wave Infrared (1-2 μm)
UNC	University of North Carolina
USAF	United States Air Force
VIS	Visible (0.4-0.7 μm)

APPENDIX A - INTELLECTUAL PROPERTY RESULTING FROM THIS PROGRAM

We are currently investigating the possibility of patenting the night-vision enhancement techniques discussed in this report. Under immediate consideration are the ASTA noise reduction algorithm and its associated tone mapping technology. Because the details of this algorithm were publicly disclosed as of August 1st in written form and August 3rd in an oral presentation at the SIGGRAPH technical conference in Los Angeles, we have less than one year to pursue a U.S. patent. We believe this technology has wide-reaching uses in addition to night-systems, such as for scientific video enhancement to film restoration, and deserves patent consideration.

The University of North Carolina's Office of Technology Development handles all internal patent requests for the science departments at UNC Chapel Hill. After the internal definition and discussion of our invention, their team begins the initial patent prior-art search and also decides if a patent is an economically reasonable option. The Office of Technology Development will also handle any licensing issues pertaining to the technology should intellectual property rights be obtained.

APPENDIX B - PUBLICATIONS AND PRESENTATIONS

B1. Publications

1. "Video Enhancement Using Per-Pixel Virtual Exposures," Bennett, E.P., and McMillan, L., ACM Transactions on Graphics, 24, 3, 845-852, (01-08-05).

B2. Presentations

1. "A Virtual Exposure Camera Model," McMillan, L., 2005 Symposium on Computational Photography and Video, Massachusetts Institute of Technology Computer Science and Artificial Intelligence Laboratory, Cambridge, MA, 23-05-2005.

2. "Video Enhancement Using Per-Pixel Virtual Exposures," Bennett, E.P., ACM SIGGRAPH (Association for Computing Machinery Special Interest Group on Computer Graphics and Interactive Techniques), Los Angeles Convention Center, Los Angeles, CA, 08-03-2005.

APPENDIX C - PROFESSIONAL PERSONNEL ASSOCIATED WITH THIS PROGRAM

Persons who contributed to technical work on this effort by role are listed below.

Company	Name	Role
University of North Carolina		
	Prof. Leonard McMillan	Principal Investigator
	Prof. Wei Wang	Co-Principal Investigator
	Mr. Eric Bennett	Graduate Student for Prof. Leonard McMillan
	Mr. Jingdan Zhang	Graduate Student for Prof. Leonard McMillan
	Mr. John Mason	Graduate Student for Prof. Leonard McMillan
	Ms. Tynia Yang	Graduate Student for Prof. Leonard McMillan and Prof. Wei Wang
	Mr. Guodong Liu	Graduate Student for Prof. Wei Wang

APPENDIX D - REFERENCES

ACOSTA-SERAFINI, P. M., MASAKI, I., and SODINI, C.G. 2004, Predictive Multiple Sampling Algorithm with Overlapping Integration Intervals for Linear Wide Dynamic Range Integrating Image Sensors. IEEE Transactions on Intelligent Transportation Systems, 5, 1, 33-41.

BARASH, D. 2002. A Fundamental Relationship Between Bilateral Filtering, Adaptive Smoothing, and the Nonlinear Diffusion Equation. Transactions on Pattern Matching and Machine Learning, 24, 6, 844-847.

BENNETT, E.P. and MCMILLAN, L. 2003. Proscenium: A Framework for Spatio-Temporal Video Editing. In Proceedings of ACM Multimedia 2003, 177-183.

BENNETT, E.P. and MCMILLAN, L. 2005. Video Enhancement Using Per-Pixel Virtual Exposures. ACM Transactions on Graphics, 24(3), 845-852.

BIDERMANN, W., EL GAMAL, A., EWEDEMI, S., REYNERI, J., TIAN, H., WILE, D., and YANG, D., 2003. A .18μm High Dynamic Range NTSC/PAL Imaging System-on-Chip with Embedded DRAM Frame Buffer, In Proceedings of the IEEE International Solid-State Circuits Conference, 212-213.

BOOMGAARD R. v. d., and WEIJER, J. v. d. 2002. On the Equivalence of Local-Mode Finding, Robust Estimation and Mean Shift Analysis As Used In Early Vision Tasks. In Proceedings of the International Conference on Pattern Recognition, 927-930.

CHEN, B. H., PAU, L. F., and WANG, P. S. P. 1998. The Handbook of Pattern Recognition and Computer Vision (2nd Edition), World Scientific Publishing Co., 207-248.

CHOUDHURY, P. and TUMBLIN, J. 2003. The Trilateral Filter for High Contrast Images and Meshes. In Proceedings of the Eurographics Symposium on Rendering 2003. 1-11.

COHEN, M., COLBURN, A., and DRUCKER, S. 2003. Image Stacks. Microsoft Research Technical Report, MSR-TR-2003-40.

COMANICIU, D. and MEER, P. 2002 Mean Shift: A Robust Approach Toward Feature Space Analysis. IEEE Trans. Pattern Analysis Machine Intelligence, 24(5), 603-619.

DEBEVEC, P. E. and MALIK, J. 1997. Recovering High Dynamic Range Radiance Maps from Photographs. In Proceedings of ACM SIGGRAPH 1997, ACM SIGGRAPH / Addison Wesley, Computer Graphics Proceedings, Annual Conference Series, 369-378.

DRAGO, F., MYSZKOWSKI, K., ANNEN, T., and CHIBA, N. 2003. Adaptive Logarithmic Mapping for Displaying High Contrast Scenes. In Proceedings of EUROGRAPHICS 2003, 22, 3, 419-426.

DUBOIS, E. and SABRI, S., 1984. Noise Reduction in Image Sequences Using Motion-Compensated Temporal Filtering, IEEE Transactions on Communications, 32, 7, 826-831.

DURAND, F. and DORSEY, J. 2002. Fast Bilateral Filtering for the Display of High-Dynamic Range Images. ACM Transactions on Graphics, 21, 3, 257-266.

EISEMANN, E. and DURAND, F. 2004. Flash Photography Enhancement via Intrinsic Relighting. ACM Transactions on Graphics, 23, 3, 670-675.

FATTAL, R., LISCHINSKI, D., and WERMAN, M. 2002. Gradient Domain High Dynamic Range Compression. ACM Transactions on Graphics, 21, 3, 249-256.

FELZENSZWALB, P.F. and HUTTENLOCHER, D.P., 2004. Efficient Belief Propagation for Early Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

FODOR, I.K. and C. KAMATH, 2003, "Denoising through Wavelet Shrinkage: An Empirical Study," Journal of Electronic Imaging, 12,151-160.

FRANCIS, J. J. and JAGER, G. D. 2003. The Bilateral Median Filter. In Proceedings of the 14th Symposium of the Pattern Recognition Association of South Africa.

FREEMAN, W.T., PASZTOR, E.C., and CARMICHAEL, O.T., 2000 Learning Low-Level Vision. International Journal on Computer Vision, 40, 1, 25-47.

JOBSON, D. J., RAHMAN, Z.-U., and WOODELL, G. A. 1997. A Multiscale Retinex for Bridging the Gap Between Color Images and the Human Observation of Scenes. IEEE Transactions on Image Processing, 6, 7, 965-976.

JOSTSCHULTE, K., AMER, A., SCHU, M., and SCRODER, H., 1998. Perception Adaptive Temporal TV-Noise Reduction Using Contour Preserving Prefilter Techniques, IEEE Transactions on Consumer Electronics, 44, 3 (August), 1091-1096.

KANG, S. B., UYTTENDAELE, M., WINDER, S., and SZELISKI, R. 2003. High Dynamic Range Video, ACM Transactions on Graphics, 22, 3, 319-325.

LEE, S. H. and KANG, M. G. 1998. Spatio-Temporal Video Filtering Algorithm based on 3-D Anisotropic Diffusion Equation. In Proceedings of the International Conference on Image Processing, 98, 2, 447-450.

LEVIN, A., LISCHINSKI, D. and WEISS, Y. 2004. Colorization Using Optimization. ACM Transactions on Graphics, Aug 2004.

LIU, X., and EL GAMAL, A., 2003. Synthesis of High Dynamic Range Motion Blur Free Image From Multiple Captures. IEEE Transactions on Circuits and Systems, Fundamental Theory and Applications, 50, 4, 530-539.

MALFAIT, M. and ROOSE, D. 199.7 Wavelet-Based Image Denoising Using a Markov Random Field a Priori Model. IEEE Transactions on Image Processing, 6, 549–565.

MALIK, J. and PERONA, P. 1990. Preattentive texture discrimination with early vision mechanisms. Journal of the Optical Society of America, 7, 5, 923–932.

MALIK, J., BELONGIE, S., LEUNG, T., and SHI, J. 2001. Contour and Texture Analysis for Image Segmentation. International Journal of Computer Vision, 43, 7-27.

NASON, G. and SILVERMAN, B. 1995. The Stationary Wavelet Transform and Some Statistical Applications. Wavelets and Statistics, Antoniadis and Oppenheim, 281-299.

NAYAR, S. and BRANZOI, V. 2003. Adaptive Dynamic Range Imaging: Optical Control of Pixel Exposures over Space and Time. In Proceedings of the International Conference on Computer Vision, 1-8.

NAYAR, S. and BRANZOI, V. 2004. Programmable Imaging Using a Digital Micromirror Array. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 436-443.

PATTANAIK, S. N., TUMBLIN, J., YEE, H. and GREENBERG, D. 2000. Time Dependent Visual Adaptation for Fast Realistic Image Display. In Proceedings of ACM SIGGRAPH 2000, ACM SIGGRAPH / Addison Wesley, Computer Graphics Proceedings, 47-54.

PERONA, P. and MALIK, J. 1990. Scale-Space and Edge Detection Using Anisotropic Diffusion. IEEE Transactions of Pattern Matching and Machine Intelligence, 12, 7, 629-639.

PEREZ P., GANGNET, M., and BLAKE, A. 2003. Poisson Image Editing. ACM Transactions on Graphics, 22, 3, 313-318.

PETSCHNIGG, G., AGRAWALA, M., HOPPE, H., SZELISKI, R., COHEN, M.F., and TOYAMA, K. 2004. Digital Photography with Flash and No-Flash Pairs. ACM Transactions on Graphics, 23, 3, 661-669.

RANDEN, T. and HUSOY, J. H. 1999. Filtering for Texture Classification: A Comparative Study. IEEE Transactions on Pattern Recognition and Machine Intelligence, 21(4), 291-310.

RASKAR, R., ILIE, A., and YU, J. 2004. Image Fusion for Context Enhancement and Video Surrealism. In Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering, 85-94.

REIBEL, Y., JUNG, M., BOUHIFD, M., CUNIN, B., and DRAMAN, C. 2003. CCD or CMOS Camera Noise Characteristics. In Proceedings of the European Physical Journal of Applied Physics, 75-80.

REINHARD, E., ASHIKHMIN, M., GOOCH, B., and SHIRLEY, P. 2001 Color Transfer between Images. IEEE Computer Graphics and Applications, September/October 2001, 34-40.

RUDERMAN, D.L., CRONIN, T.W., and CHIAO, C.C. 1998. Statistics of Cone Responses to Natural Images: Implications for Visual Coding. Journal of the Optical Society of America, 15, 8, 1998, 2036-2045.

SAND, P. and TELLER, S. 2004. Video Matching. ACM Transactions on Graphics, 23, 3, 592-599.

SRINIVASAN, A. and ANDERSON, J. 2003. "Fair Scheduling of Dynamic Task Systems on Multiprocessors", In Proceedings of the 11th International Workshop on Parallel and Distributed Real-time Systems.

STOCKHAM, T.G. 1972. Image Processing in the Context of a Visual Model, In Proceedings of the IEEE, 60, 828-842.

SUN, J., ZHENG, N.N., and SHUM,H.Y., 2003. Stereo matching using belief propagation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25, 7, 787–800.

TAI Y.-W., JIA, J., and TANG, C.-K., 2005. Local Color Transfer via Probabilistic Segmentation by Expectation-Maximization. In the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition , I, 747-754.

TOET, A. 2005 Colorizing Single Band Intensified Nightvision Images. Displays, 26, 1, January, 15-21.

TOMASI, C. and MANDUCHI, R. 1998. Bilateral Filtering for Gray and Color Images. In Proceedings of the International Conference on Computer Vision, 836-846.

TUMBLIN, J. and RUSHMEIER, H.E. 1993. Tone Reproduction for Realistic Images. IEEE Computer Graphics and Applications,13,6,42-48.

TUMBLIN, J. and TURK, G. 1999. LCIS: A boundary hierarchy for detail preserving contrast reductions. In Proceedings of SIGGRAPH 1999, 83-90.

VAN DE WOUWER, G. SCHEUNDERS, P., and VAN DYCK, D. 1999. Statistical Texture Characterization from Discrete Wavelet Representations. IEEE Transactions on Image Processing, 8, 592-598.

VARGA, J.T. 1999. Evaluation of Operator Performance Using True Color and Artificial Color in Natural Scene Perception (Report AD-A363036), Naval Postgraduate School, Monterey, CA.

WARD, G. 1991. Real Pixels. Graphics Gems II. Academic Press. 80-83.

WELSH, T., ASHIKHMIN, M., AND MUELLER, K. 2002. Transferring color to greyscale images. ACM Transactions on Graphics 21, 3, 277-280.

YEE, H., PATTANAIK, S, and GREENBERG, D. P. 2001. Spatio-Temporal Sensitivity and Visual Attention for Efficient Rendering of Dynamic Environments. ACM Transactions on Graphics, 20, 1, 39-65.

YEDIDIA, J.S., FREEMAN, W.T., WEISS, Y. 2003. Understanding Belief Propagation and Its Generalizations, Exploring Artificial Intelligence in the New Millennium, ISBN 1558608117, Chapter 8, 239-236.

ZHANG, Y. 1997. Evaluation and Comparison of Different Segmentation Algorithms. Pattern Recognition Letters, 18(1997) 963-974.

APPENDIX E - FILTERSHOP MANUAL

FilterShop
Image and Video Enhancement Application
October 2005

Introduction:
FilterShop is an application that encapsulates the algorithms and techniques developed for night-vision at UNC-Chapel Hill. It allows these tools to be used within a graphical user interface similar to popular image editing applications. Along with the new techniques, a complement of supporting tools and pre-existing algorithms are included for manipulation and benchmarking.

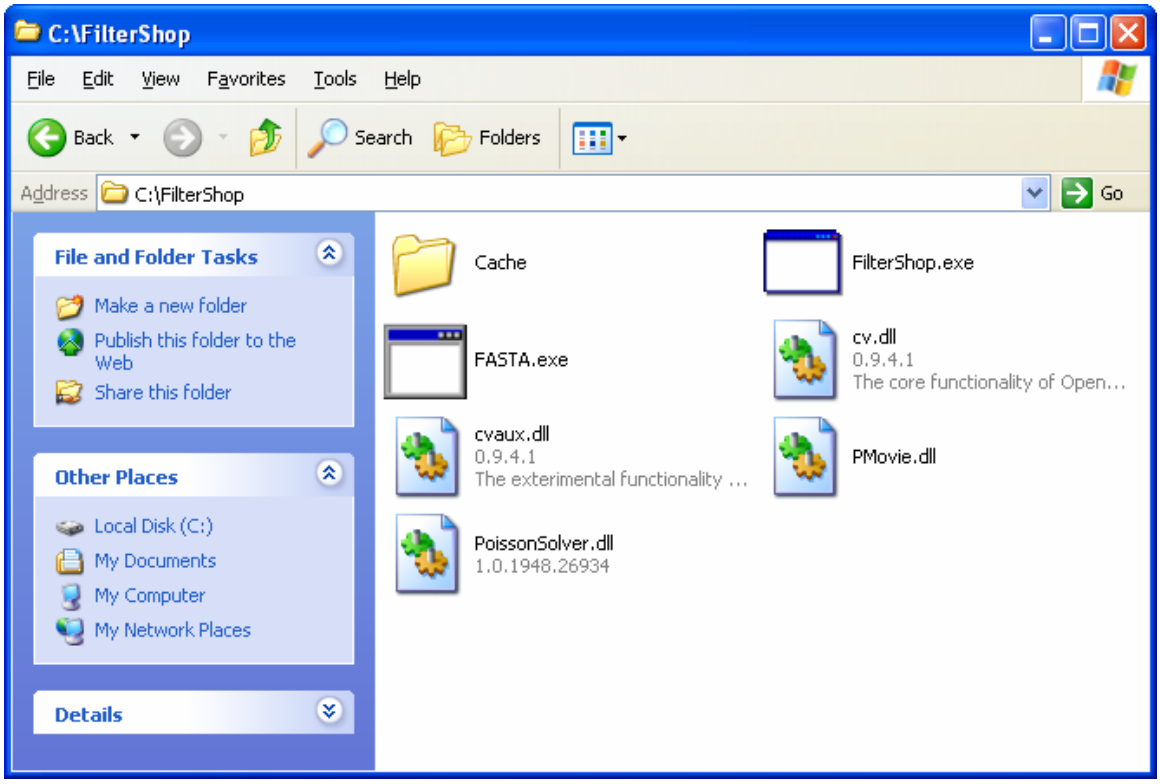
Major Features:
- ASTA Noise Reduction
- fASTA High-Speed Noise Reduction
- Learning-based Colorization
- Iterative and Multi-Grid Poisson System Solvers
- Belief Propagation System Solvers
- Multiple color space representations
- Library of Noise Reduction algorithms

Supported Image I/O Formats:
- JPEG (.JPG)
- PNG (.PNG)
- TIFF (.TIF)
- BMP (.BMP)

Supported Video I/O Formats:
- Video for Windows (.AVI) 8-bit pixels
- FilterShop Data (.FSD) Floating point pixels

Minimum System Specifications:
- Microsoft Windows XP or later
- Microsoft .NET Framework v1.1 or later
- Pentium IV 1.0GHz or faster
- 256 MB of RAM (512 or 1024 MB Suggested)
- 1 GB of free hard disk space (2 GB Suggested)

Installation Guide:
Please place the application and associated files in the C:\FilterShop folder. Verify that the C:\FilterShop\Cache folder exists. Without this folder, FilterShop cannot process video.



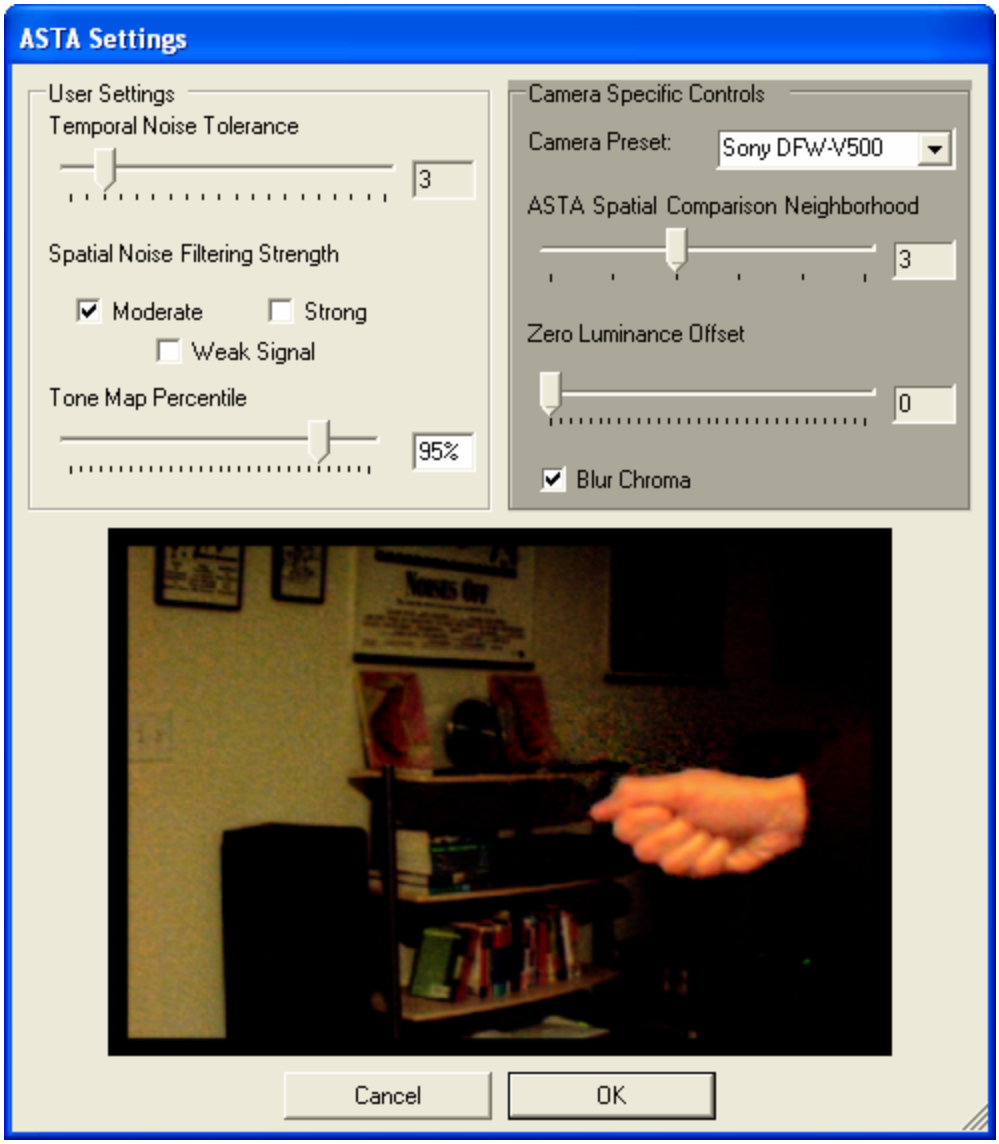
Errata:

If FilterShop demonstrates erratic behavior (such as after a system crash) please clear the contents of C:\FilterShop\Cache

The fASTA.exe application can be called directly without using the FilterShop interface. Running fASTA.exe at the command prompt without any arguments will display a brief usage guide. However, all typical use of the application can be accomplished without having to use this interface.

Getting Started Tutorials

Example: *Using High-Speed ASTA to enhance video*
1) Choose Menu Item “fASTA / Full fASTA”
2) Select the .AVI file of the video you wish to enhance and press “Open”
3) Adjust the parameters and see the single frame update in the lower window.
4) When you are happy with the result, press OK.
5) When the video is done processing, choose “File / Save” or “File / Export / AVI” to save the result to disk.



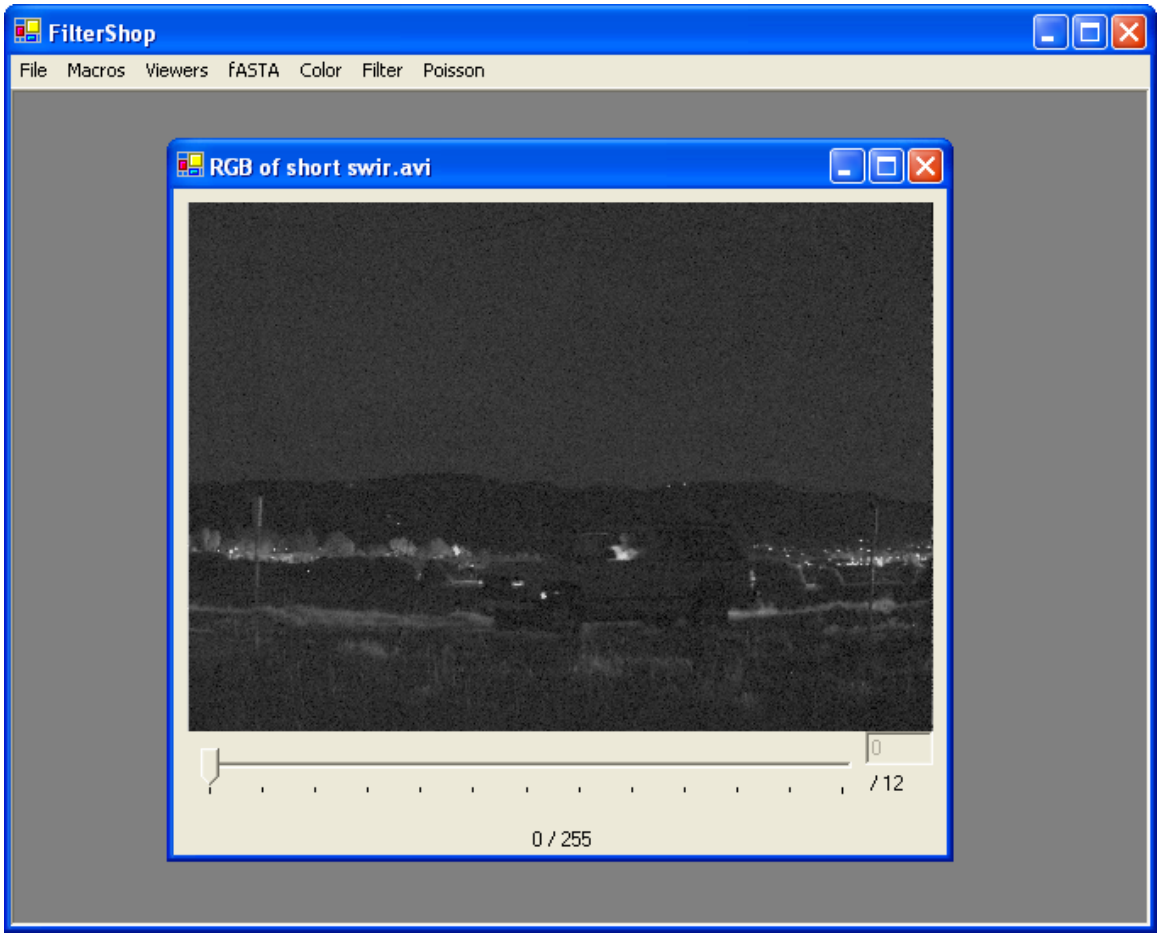
Example: *Use an example image to colorize a gray image or video*
1) Choose “File / Open” to open both the original image and the image or video to colorize.
2) Select the window with the image or video you wish to colorize.
3) Choose “Filter / Colorize”
4) Select the example color image from the list.
5) When the new colorized image or video appears, use “File / Save”, “File / Export / BMP”, or “File / Export / AVI” to save the file to disk.

Example: *Subtract a single dark image from a video*
1) Choose “File / Open” to open both the dark image and the video you wish to subtract it from.
2) Select the window with the dark image
3) Choose “Filter / Structural / Match Frame Count”
4) Select the video from the list
5) Select the window with the original video
6) Choose “Filter / Subtract”
7) Choose the new, multi-frame version of the original movie from the list.
8) If the new video is too dark, choose “Filter / Functional / Stretch To Range” to stretch the dynamic range
9) Save the new video with either “File / Save” or “File / Export / AVI”

Example: *Use the gradient field of one image and the boundary conditions of another*
1) Open both of the images using “File / Open”
2) Use “Color / Manipulate YUV / Extract YUV” on both images.
3) Use the “Filter / Functional / Shift Zero Point” filter on both images to offset them each by 1 (removing the zero log case)
4) Select one video, and use “Poisson / Integration Field Tools / Find Gradient”
5) Choose “Poisson / Gradient Integration / Multi-Grid / Rectangular”
6) Select the image from the list with the boundary conditions and press “OK”
7) Select the gradient field you just created from the list and press “OK”
8) To restore the original color, select the original image whose chrominance you want to keep, and select “Color / Manipulate YUV / Replace YUV”
9) Choose the newly integrated luminance field
10) Save the new image using either “File / Save” or “File / Export / BMP”

Reference Guide

The Main Window:



- The view shows a still image or movie frame with values clamped at 0 and 255.
- The slider changes the frame number of the movie shown in the view
- The frame counter shows both the current frame number and the total frames
- The text at the bottom shows the "minimum/maximum" value in all channels

File Menu:

- Open

Opens .bmp, .jpg, .tif, .png, .avi, and .fsd files. FSD files are FilterShop's internal floating-point image and video format.

- Save

Saves an FSD file of the current window. If it is a movie, the file will contain all of the frames

- Export : AVI

Exports a 30fps AVI of the current window. A dialog will prompt which installed AVI codec will be used.

- Export : BMP

Exports a BMP of the frame currently shown in the selected window.

Macros:

Macro functionality is limited to basic image manipulation functions. More complex techniques, such as ASTA and Poisson interpolation, are not recorded to or played back from macro files.

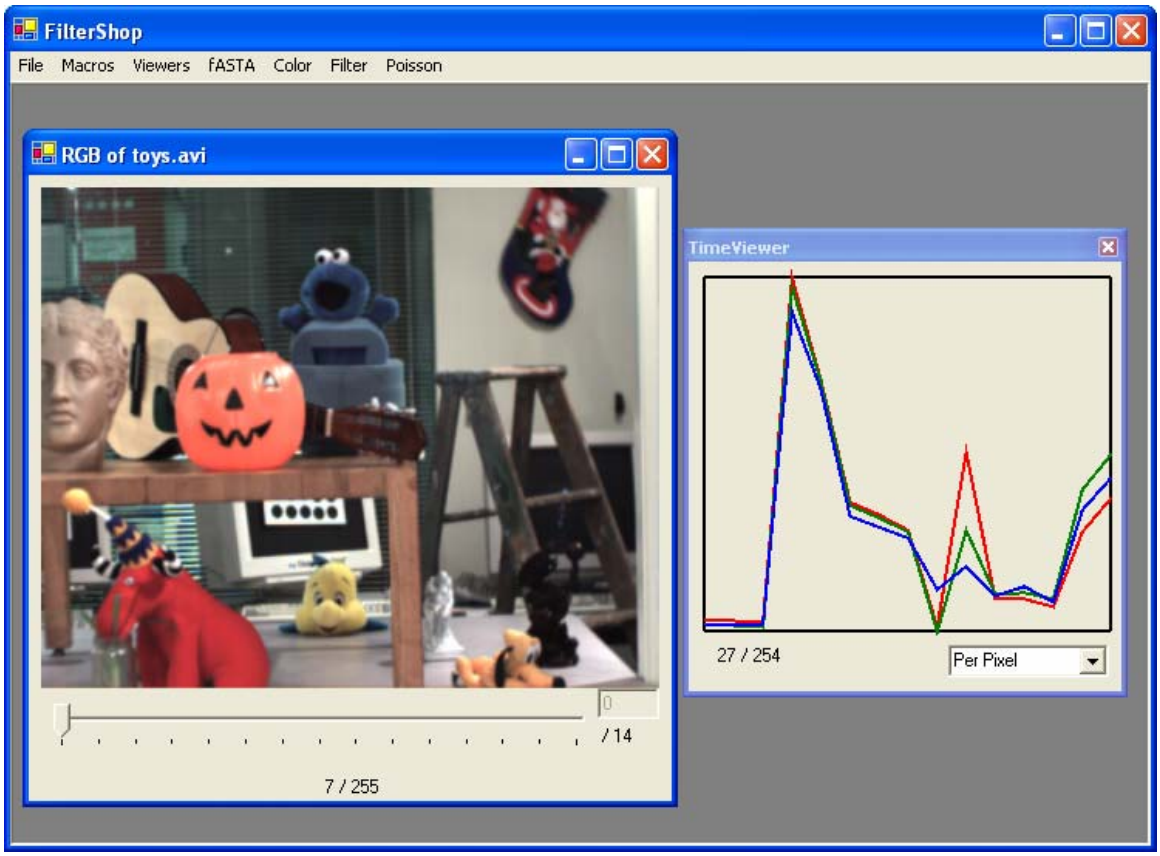
- Play
  Plays back a recorded series of actions
- Record / Stop
  Prompts for a file name and records user actions. This menu option reads "Stop" during the process, and selected this menu option will save the script to a file.
- Run My Script
  Unused

Viewers:

- Temporal

Opens a viewing window that shows a 1D plot of pixel intensity through time for the pixel located under the mouse cursor in the active window. Each component color channel of RGB is shown as its own color line.

The vertical scale can be changed to accentuate different factors. Each sample can be mapped to maximally zoom into each 1D plot, each sample can be mapped to between 0-255, or each sample can be mapped between the movie's minimum and maximum of all frames. This mapping can be adjusted via the pull-down menu at the bottom-right of the temporal viewer. If something other than the 0-255 range is selected, the range will be displayed as a "Minimum/Maximum" label at the bottom of the window.



Errata:

Do not use this feature on movies longer than 40 frames, or else extreme disk thrashing will occur in an attempt to read full cached frames from disk to retrieve only a single pixel from each frame.

fASTA:

Given a single video, fASTA produces a cleaned and tone-mapped version. The primary control is the temporal noise tolerances. Too much temporal tolerance, and ghosting results between frames. Too little spatial tolerance and moving objects appear grainy. Strong filtering engages use of a median filter. The tone map percentile slider allows you to avoid undersaturating pixels in images with large dynamic ranges.

By using the fASTA form of the ASTA algorithm, it is possible to give feedback in a matter of seconds regarding the output quality of the settings. Once settings have been chosen, the fASTA algorithm will execute inside a command prompt. Upon completion, the new movie will appear in a new window in FilterShop.

The camera specific settings usually do not need to be tweaked, but can be if desired. The half-kernel size of the "distance measure" for neighborhood differences can be adjusted to overcome heavy noise. Some cameras do not have a true 0 value, so an offset can be specified. Finally, blurring of chrominance (usually suggested) can be turned off.

- Full fASTA

This performs the ASTA pipeline, including noise reduction and tone-mapping. Choose the .AVI file to process from the dialog, and then configure using the second dialog.

- ASTA Only

Same as above, but bypasses tone-mapping, resulting in just noise reduction.

Color:

Currently, FilterShop supports the RGB, YUV, Flash Luminance, and YUV Luminance color spaces. Although other color spaces are listed in menus (e.g. Vector), they are unsupported as of the October 2005 release.

- Convert to X

Converts the internal color representation to the chosen new color space. The current color mode will be grayed out in the sub-menu of choices.

- Fuse to X

Creates a new image sequence based on single color channels from already open movies. Choose the target color space from the sub-menu.

- Substitute Channel:

Changes one of the channels of the currently selected window to that of a different window. A dialog will prompt to choose the video channel to insert.

- Extract Channel:

Makes a grayscale movie from a single channel of the current window.

- Manipulate YUV : Extract YUV Lum

Creates a movie of the luminance of the current window in YUV Space.

- Manipulate YUV : Replace YUV Lum

Replace the Y channel of the video in the selected window (regardless of color space) with that of a different video selected via a dialog.

- Extract FLASH Lum

Calculates the Luminance using the [Durand 02] method (Preferred).

Filter:

- Spatial : Bilateral

Performs a standard Bilateral filter on each frame of a video. The spatial and intensity Gaussian falloffs are variable, along with the half-kernel size.

- Spatial : Trilateral

Performs the "Trilateral Filter". This is very slow and is based off of code from the original Trilateral paper authors.

- Spatial : Gaussian

Performs a Gaussian blur with a given intensity falloff and kernel size.

- Spatial : Maximum

The filter prompts for a kernel half-size. It then picks the largest value within that kernel centered at each pixel.

- Spatial : Median

Performs an n+1 by n+1 median kernel to all pixels. The value entered is n/2.

- Spatial : Bilateral-Median

Presents the Bilateral interface, than a dialog to enter a median scalar. It then applies the Median Centered Bilateral as described in the ASTA section.

- Spatial : Absolute Gaussian

Performs a Gaussian over the absolute value of the video.

- Temporal : Bilateral

Performs a 1-D bilateral through time. The temporal Gaussian falloffs (for intensity difference and spatial distance) and a half-kernel size are required.

- Temporal : Integral

Adds up a given number of frames before and after each frame. No normalizing divide is performed afterwards.

- Temporal : Gaussian

Performs a per-pixel Gaussian blur through time.

- Temporal : 3D Bilateral

Performs a spherical bilateral noise filter. Does not work well, as discussed in the paper.

- Functional : Floor / Ceiling

Clamps the video to a new minimum and a new maximum.

- Functional : Stretch to Range

Specify a new minimum and a new maximum for the video.

- Functional : Shift Zero Point

Applies a scalar offset to all values. All negative results will be clamped at 0.

- Functional : Add

Adds a second video to the video in the current window.

- Functional : Subtract

Subtracts a second video from the video in the current window.

- Functional : Scale

Specify a scalar multiplier that will be applied to all pixels in all channels.

- Functional : Modulate

Performs a per-pixel multiply between a second video and the video in the current

window.
- Functional : Divide
    Performs a per-pixel divide between a second video and the video in the current window.
- Functional : Log
    Takes the natural log of each pixel in each channel. It is suggested that all values should be greater than 0 to avoid issues.
- Functional : Exp
    Takes the inverse natural log of each pixel in each channel.
- Functional : Zero Abs Range
    Unsupported
- Functional : Flip
    Subtracts the video's maximum value from all values
- Functional : Abs
    Takes the absolute value of all values
- Functional : Average
    Averages all frames to create a single arithmetic-mean output frame.

- Structural : Copy
    Creates an exact copy of the movie in the selected window.
- Structural : Extract Still
    Opens a new window containing just the single frame viewed in the selected window.
- Structural : Quarter Size
    Reduces the spatial resolution of the movie by half via bilinear interpolation.
- Structural : Match Frame Count
    Adds repeated copies of the final frame of a movie to match the duration of a second movie chosen in a pop-up dialog.

- HDR : Build Dark HDR
    Unsupported
- HDR : 10-50-500
    Unsupported
- HDR : Gamma Correct
    Applies a 1/gamma global tone mapping.

- Histogram : Lum Histogram
    Creates a grayscale histogram movie of the selected window's movie
- Histogram : Color Histogram
    Creates a multichannel histogram of each of the selected movie's channels
- Histogram : Fit to Image
    Attempts to approximate the "look" of the video in the current window by performing least squares fit to a second video chosen in a dialog.

- Field Analysis : Gradient
    Creates a gradient field. The field can be a forward difference or a central difference. The result is either a total gradient magnitude, or a multi-channel gradient with the 0 channel

being X, 1 = Y, and 2 = Z.
- Field Analysis : Contrast
    Creates a (B+A)/(B-A) contrast field that is NOT compatible with the Poisson solver.
- Field Analysis : Score
    Generates a per-pixel local neighborhood "distance measure" between the current frame and all other frames. The result is useful for tweaking ASTA settings.

- ASTA
    Performs ASTA algorithm as published. These commands are now deprecated. Please use the fASTA algorithm described above.
- Interactive ASTA
    Deprecated

- ASTA Tone Map
    Performs just the tone mapping stage of the ASTA pipeline. This command is now deprecated. Please use the fASTA algorithm described above.
- Interactive ASTA Tone Map
    Deprecated

- Colorization
    Under Development

Poisson:

- Integration Field Tools : Find Gradient
  Creates a Poisson compatible gradient field
- Integration Field Tools : Find Contrast
  Creates a Poisson compatible contrast field
- Integration Field Tools : Mix with Mask
  Given a 0/255 mask, mixes two fields and two images together.
- Integration Field Tools : Attenuate
  Performs a "Gradient Domain High Dynamic Range Compression"-style algorithm.

- Other Tools : Make Gray Field
  Makes an all gray (127) version of the video
- Other Tools : Make Simple Mask
  Makes a white (255) version of the video with a 5 pixel black border
- Other Tools : Guess Integration
  Using a gradient field, attempts naïve integration to create a plausible starting condition.

- Gradient Integration : Gauss-Seidel
  Performs an iterative solution to gradient integration assuming border boundary conditions. Requires a starting image and a gradient field. This will run until the user clicks on the iterating image.
- Gradient Integration : Multi-Grid : Rectangular
  Performs a single pass multi-grid solver that maintains the value of the pixels on the border of the starting image. Requires a starting image and a gradient field.
- Gradient Integration : Multi-Grid : Non-Rectangular
  Performs a single pass multi-grid solver that can use an arbitrarily shaped boundary condition. A starting image, a gradient field, and a mask are required. White areas of the mask will be integrated, while black areas will be held as boundary conditions.

Same as above, but solved using a multiplicative solution to a contrast field integration problem.

- Contrast Integration * : Gauss-Seidel
- Contrast Integration * : Multi-Grid : Rectangular
- Contrast Integration * : Multi-Grid : Non-Rectangular

Same as above, but solved using an additive solution to a contrast field integration problem.

- Contrast Integration + : Gauss-Seidel
- Contrast Integration + : Multi-Grid : Rectangular
- Contrast Integration + : Multi-Grid : Non-Rectangular